BlockChain: A distributed solution to automotive security and privacy

Ali Dorri, Marco Steger, Salil S. Kanhere, and Raja Jurdak

Future smart vehicles will be part of the Internet of Things to offer beneficial development opportunities for both end users as well as the automotive industry. This will potentially expose smart vehicles to a range of security and privacy threats such as tracking or hijacking a vehicle while driving. A comprehensive security architecture for automotive systems is required to allow the development of new services while protecting the vehicles from attacks and ensuring the privacy of the end users.

In this paper we argue that BlockChain (BC), a disruptive technology that has found many applications from cryptocurrency to smart contracts, is a potential solution to automotive security and privacy challenges. We propose a BC-based architecture to protect the privacy of the users and to increase the security of the vehicular eco-system. Wireless remote software updates and other emerging services in the automotive world such as dynamic vehicle insurance fees, are used to illustrate the utilization of the proposed security architecture. We also provide discussions on the security of the architecture against important attacks.

Introduction

Modern vehicles are increasingly connected to roadside infrastructure e.g. traffic management systems, to other vehicles in close proximity, and also more generally to the Internet, thus making future vehicles part of the Internet of Things (IoT).

A smart vehicle requires a Wireless Vehicle Interface (WVI) to interconnect with the IoT as shown in Figure 1. On behalf of their driver, smart vehicles can use their WVI to exchange data with other entities e.g. other vehicles or the OEM (i.e. the car manufacturer) to provide personalized services for the vehicle owner. This increased degree of interconnectivity of vehicles can be critical, as the WVI can be exploited by an attacker to endanger the integrity of the vehicle and the privacy of the owner. The exchanged information can include sensitive data and can thus open up new privacy challenges for smart vehicles.

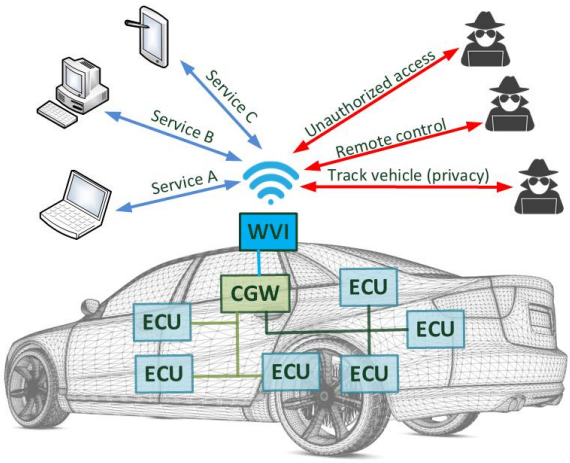

Figure 1 *A future smart vehicle utilizing a wireless vehicle interface (WVI) to interconnect the vehicle and its vehicular bus systems to the Internet.*

The white-hat hackers Miller and Valasek presented a sophisticated attack on a Jeep Cherokee using the wireless interface of the infotainment system and were able to remotely control the vehicle's core functions [1]. This attack is one among many attacks showing that the current security systems for smart vehicles are vulnerable and need to be improved.

Novel automotive security architectures will have to address the following requirements and challenges to satisfy the needs of future services of smart vehicles [2], [3]:

- Scalability: a scalable architecture is required as the vehicular ecosystem consists of a large number of vehicles, each fitted with numerous electronic control units;
- Safety: future vehicles will have an increasing number of autonomous driving functions. A malfunction due to a security breach (e.g., by installing malicious SW) could lead to serious accidents thereby endangering the safety of the passengers of the vehicle and also of other road users in its close proximity. A security architecture for smart vehicles must be able to protect against such threats;

- Centralization: a centralized security solution is not applicable for future smart vehicle networks, where all vehicles can connect to each other, due to single-point-of-failure and low scalability. Therefore, a decentralized solution is preferred.
- Maintainability: automotive security architectures must address maintainability (i.e., provide SW as well as HW support for vehicles for at least ten years) and extendibility (i.e., allowing to add new SW and HW features to a vehicle) once the vehicle is already out in the field as a vehicle is typical used for many years.

Research is currently focused on securing in-vehicle communication and smart bus gateways [5] by monitoring the packets transferred to and within the vehicle (i.e., intrusion detection). However, these methods do not provide any security mechanism for situations where vehicles are connected to the Internet.

In this article we argue that BlockChain (BC), a distributed secure and private ledger of blocks, provides a potential solution to address automotive security and privacy challenges. BC was first introduced in Bitcoin [6], the first cryptocurrency system allowing users to exchange coins in a distributed and anonymous manner. BC, as shown in Figure 2, contains blocks of transactions that are chained together using the hash of previous block that makes the BC immutable in a way that changing one block would require to change headers of the subsequent blocks. Network participants collaboratively process the BC by verifying and storing new transactions into blocks. A transaction is verified by examining its signature and checking the existence of the previous transaction in the same ledger. When the size of the transactions in the running pool, i.e. collection of received transactions that are not appended to the BC, of a node reaches a pre-defined size (known as block size), the node generates a new block by executing a consensus algorithm. The consensus algorithm differs based on the type of the BC used in the network. Proof of Work (POW) [6] and Proof of Stake (POS) [7] are two popular consensus algorithms. POW demands high computational resources while POS demands both memory and computational resources for solving a hard-to-solve easy-to-verify puzzle. In BC, users are known by a Public Key (PK) which can change per transaction. This feature introduces a high level of privacy as each user has many anonymous identities. BC presents an elegant way to achieve decentralization while also ensuring security and privacy over an untrusted network, which has resulted in its adaptation for non-monitory applications e.g. smart contracts [7] and storing data [8]. In our previous research, we optimized the BC for lightweight IoT devices by introducing a new BC consensus algorithm used for generating and processing the BC [10,11]. Our instantiation of BC maintains the security and privacy of the BC while removing the demand for solving a resource consuming puzzle, thus increasing the BC scalability and throughput. We separated the data and BC transactions flow and utilized the Open Shortest Path First (OSPF) routing protocol to route data packets directly to the requester. These modifications decrease the delay from end user perspective, and also reduce packet overheads in the network.

This paper's contribution is to design a BC-based architecture for automotive security and privacy. To the best of authors' knowledge this is the first security architecture based on BC in the automotive field. In our architecture OEMs, SW update providers, and smart vehicles jointly generate an overlay where they can share data in a distributed way as shown in Figure 3. The overlay is clustered and then Cluster Heads -known as Overlay Block Managers (OBM)- manage a public BC that stores transactions (i.e. communications) between overlay nodes. A vehicle can join to the overlay either directly or via smart buildings. The smart building utilizes a private network (e.g., a smart home) and has a private, centrally-managed BC, known as Immutable Ledger (IL), containing all transactions related to the building. The local IL is linked to the overlay BC by storing its hash that contains the hash of a local storage in the public BC. The local storage, especially in smart homes, is used to store privacy-sensitive data, e.g. vehicle location, to protect the vehicle owner's privacy. Benefiting from the BC the vehicle owner can prove that the data in this local storage has not changed since generation. Our architecture supports static as well as mobile vehicles, however, mobile vehicles might face small delays before receiving services for the first

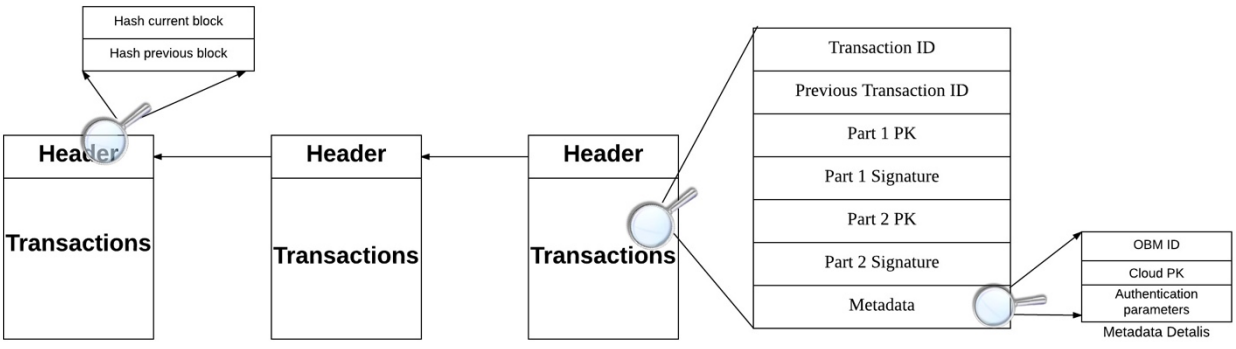

Figure 2 *The structure of the BC*

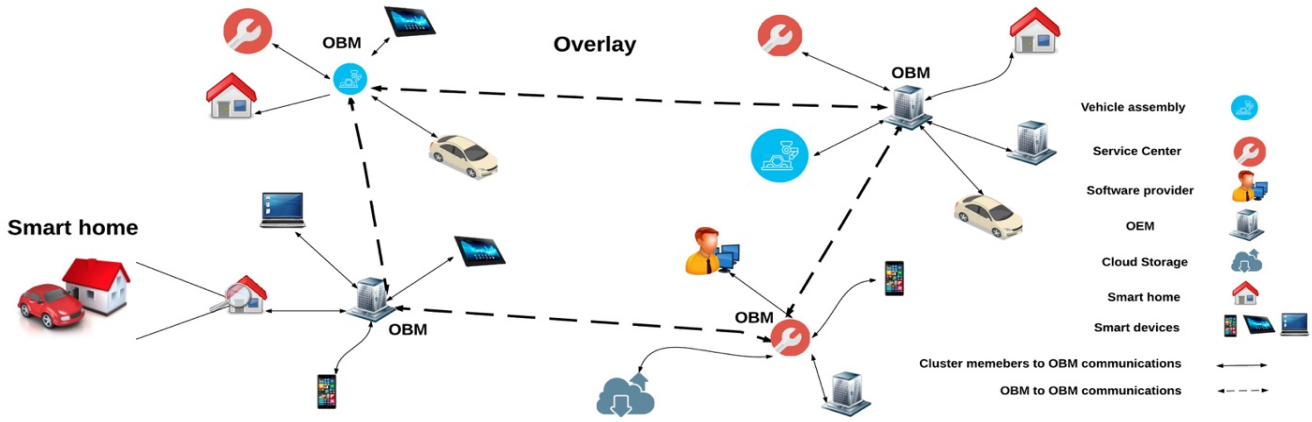

Figure 3 *An overview of the proposed architecture.*

time in a new cluster.

We next discuss our architecture and subsequently describe different automotive use cases – Wireless Remote SW Updates (WRSU), dynamic insurance fees, smart charging service, and car sharing services – to demonstrate the generality of the proposed architecture.

BlockChain-Based Architecture

In this section, we discuss the details of the proposed BC-based architecture for automotive security and privacy. The main part of our architecture is the overlay network where a public BC is managed by nodes.

The overlay, as shown in Figure 3, is comprised of different nodes including smart vehicles, smart buildings (that can be smart homes or service centers), OEMs, vehicle assembly lines, SW providers, cloud storages, and user's mobile devices such as smartphones, laptops, or tablets. Each smart building is centrally managed by a trusted node known as Local Block Manager (LBM). This LBM is integrated in the building's Internet gateway or realized as stand-alone device (e.g. f-secure [9]) and connects the smart building to the overlay network, as shown in Figure 4. LBM maintains a local private Immutable Ledger (IL) that has the same functionality as the BC but is managed centrally. The entire traffic to and from the building is routed through the LBM.

In the proposed architecture, a vehicle can connect to the overlay either directly or through the smart building. A vehicle is part of a smart building while it is within the range of the LBM (e.g., while it is parked at home). Each smart building has a secure local storage. In a service center this storage will be used to collect vehicles-specific data (e.g., the date of the last vehicle maintenance) as well as information required by the mechanics to maintain the vehicle (e.g., the latest available SW updates and repair instructions). In a smart home, vehicle-related data, user data (e.g., coming from smart devices such as smartphones), and data coming from local sensors and actuators can be collected in the local storage.

A smart vehicle can also directly connect to the overlay network. Its WVI will act as LBM and handle the data exchange with other overlay nodes. The vehicle will also offer a local storage to store vehicle-specific data.

There can be different types of cloud storages in the overlay. The first category is managed by the OEM and/or SW update provider and could be used to store latest SW updates. The second group encompasses storage which vehicle owners use for storing vehicular data to receive specific services, e.g. cloud storage provided by vehicle insurance companies. For the second category strong but anonymous user authentication is important to on the one hand protect the privacy of the user and on the other hand guarantee that the cloud provides service to users as per their service contract. To store data, the vehicle owner provides its authentication metrics (block-number and hash of previous data) to anonymously authenticate itself to the cloud. The authentication metrics are used to locate data in the cloud storage and if so, the user authentication is passed. More details are presented in [10].

The overlay is a clustered network and its clusters are maintained by Cluster Heads (CH). The CHs also manage the public BC by verifying and storing transactions and blocks and are hence also referred to as Overlay Block Manager (OBM). The OBMs are chosen by the cluster members using methods

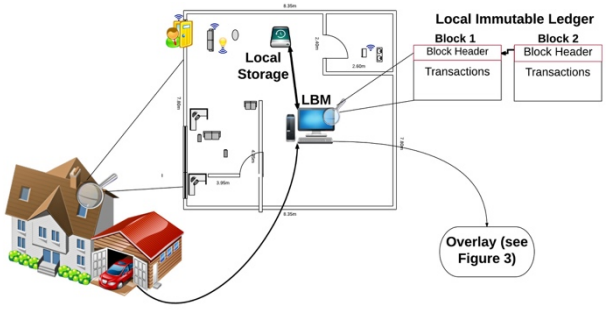

Figure 4 *A smart home managed by the Local Block Manager (LBM).*

such as described in [11]. The cluster members can elect a new CH in case an OBM fails or is compromised. In the overlay there are two types of transactions, namely:

- Single signature transactions: these transactions can be generated by overlay nodes and require the transaction generator's signature to be treated as valid.
- Multisig transactions: these transactions require two overlay node signatures to be treated as a valid transaction. The structure of this transaction is shown in Figure 2. The first field is the transaction ID that is the hash of the transaction. The second field is a pointer the previous transaction of the same overlay node. These are followed by the signature and PK of the transaction participants. The last field contains the metadata that is used by the overlay nodes to locate the cloud storage where data is stored.

Each OBM has two lists of keys to decide if a multisig transaction should be forwarded to its cluster members or to other OBMs. These lists are updated by the cluster members (i.e., overlay nodes) to allow other overlay nodes to access them. The OBM forwards a received transaction to its cluster members if i) the transaction *Part 1 PK* matches a key in the "part 1" key list, or ii) its *Part 2 PK* matches a key in the "part 2" key list. Otherwise, the OBM would broadcast the transaction to other OBMs.

In this paper, we assume that key overlay nodes including SW providers, OEMs, OBMs, and cloud storage always have a unique PK used for generating update transactions or new blocks and other nodes in the overlay are aware of their PK.

Overlay transactions are broadcast and verified by the OBMs. An OBM verifies a transaction by verifying the signature of the transaction participants with their PK. Additionally, the OBM verifies if the previous transaction mentioned in the second field of each transaction exists in the public BC and is valid. OBMs generate new blocks and broadcast them to other OBMs. On receiving a new block, an OBM verifies the transactions inside the block prior to appending it to its own BC and broadcast it to other OBMs.

In our architecture, a vehicle may be static, e.g. parked in the home, or mobile. A mobile vehicle that changes its cluster can still communicate with other overlay nodes and store transactions on the public BC, but it will be inaccessible for overlay nodes as the vehicle PK is not in the keylists of the new OBM. To make itself accessible, the vehicle should store its key on the keylists of its new OBM. In case of high mobility, the delay incurred for changing keys can be further reduced by applying methods such as mobile IP to efficiently handle the handover.

Applications

In this section we discuss various applications which can leverage the propose architecture.

Remote SW updates

Wireless Remote SW Updates (WRSU) for smart vehicles is one of the most demanding and critical security challenges in the automotive industry. WRSU are required to upgrade the functionality of the Electronic Control Units (ECU) of a vehicle or to fix a bug in the SW installed on one of these ECUs. Thereby WRSU can support the entire lifecycle of a vehicle as they can be utilized during vehicle development and assembly as well as for maintenance of the vehicle in a service center [4] or remotely from home. Today, there is no WRSU (security) architecture that can achieve WRSU in an efficient and secure manner.

The entire SW update process based on our architecture is sketched in Figure 5 and described in the following. First the SW provider creates a new SW version and stores it in cloud storage (of the OEM or the SW provider) that is accessible for all overlay nodes. Then, the SW provider creates a multisig transaction (see Figure 2) and sets the *part 1* fields of the transaction with its own PK and signature. For the signature, the SW provider uses the signed hash of the stored SW binary. As the binary is stored in the cloud, the hash can be verified by other overlay nodes thereby ensuring data integrity. The PK of the OEM is written in the *part 2* field of the transaction. This PK is later used by OBMs to forward the transaction to the OEM using their list of keys. In the next step the transaction is sent to the overlay. Currently, the multisig transaction has only one signature and hence it is not treated as valid by OBMs. The OBMs will only broadcast the transaction to the network.

Once the transaction is received by the OBM of the cluster containing the concerned OEM, the OEM is notified about the transaction. Thereafter, the OEM verifies the new SW version, signs the received transaction (*part2*) and sends it to its OBM (step 5 of Figure 5). The OBM then broadcasts the transaction to other OBMs. The OBMs verify the multisig transaction by checking the signature of both the SW provider as well as the OEM using the PKs included in the transaction. In the next step the OBMs notify their cluster members about the latest available SW update.

By receiving the transaction from the OBM, the smart vehicle verifies it by matching the PK in the transaction with its vehicle's OEM key. Finally, the vehicle downloads the SW directly from the cloud storage (step 6, Figure 5) using the authentication parameters, i.e. block-number and hash, provided in the metadata field of the received multisig transaction. This data can be used to verify that the SW version was not altered since its generation by the SW provider and OEM. The verification can be done by checking the signed hash of both the SW provider and the OEM.

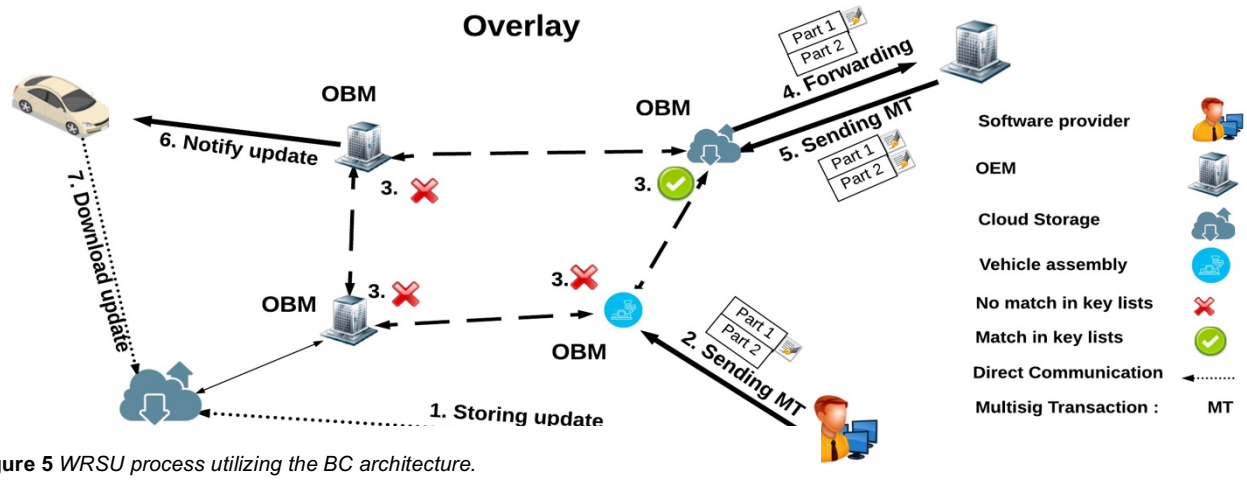

Figure 5 *WRSU process utilizing the BC architecture.*

Insurance use case

Insurance companies are beginning to offer flexible vehicle insurance fees to their responsible customers. For this, the company evaluates driving behavior using data collected from the vehicles such as braking patterns or speed. Our architecture is applicable in this situation as i) the vehicle owner has control over his personal data sent to the insurance, and ii) the insurance company can be sure that the received data is not altered by anyone.

Initially, when a car owner chooses such an insurance model, the insurance company deploys a key on the car along with cloud storage authentication parameters. These measures ensure that the data can only be read by the company, and only allow permitted costumers to store data in the cloud where a chained ledger of data is maintained.

To protect the privacy of the car owner, privacy-related information (e.g. vehicle location) that is not demanded for basic services of the insurance company, is stored in the local storage and is only accessed in case of a certain event (e.g., after an accident). Data in the local storage of either a smart building or the vehicle can be treated as trusted information as the hash of the IL, that contains the hash of the local storage, is stored in the public BC.

Electric vehicles and smart charging services

The number of electric vehicles is constantly growing. This trend increases the demand for efficient and fast vehicle charging infrastructure. By interconnecting the smart vehicle, the users' smart devices (e.g., smartphone), and the smart home the charging process will be more personalized, e.g. the smart home knows when the user typically leaves the home (based on the user's calendar). This information can be used to guarantee that the vehicle is fully charged when the user needs it while choosing the most efficient and cheapest charging strategy such as by avoiding peak load times.

The proposed security architecture allows the vehicle to exchange data with the smart home as well as with the smart devices of the user to find the best and cheapest charging strategy. The home (and vehicle) owner defines which information can be shared between these entities to protect his privacy while enabling novel services thus enriching the smart vehicle and its functionality.

Car-sharing services

Car sharing platforms and services, e.g. Uber, are growing rapidly. These highly distributed services require to interconnect smart vehicles, car-sharing service providers and of course the users of the services in a secure and reliable way. A trusted communication channel will be needed to securely exchange data including the location of the vehicle, keys to unlock the car, and payment details of the user. The proposed security architecture is eminently suitable for these services as i) the decentralized nature of BC is tailor-made for these highly distributed services, and ii) it interconnects the involved entities in a secure way while protecting the privacy of the users as well as the vehicle from unauthorized access.

Discussion

In this section we first discuss vehicle's delay tolerance, then the user privacy and finally analyze the architecture security.

The vehicle is delay tolerant in a way that it can store the vehicle parameters on the local storage when the LBM (either inside the vehicle or in smart building) is disconnected from the overlay. Once connectivity returns, the vehicle can update the public BC with the new hash of the data.

Vehicles exchange sensitive data, e.g. the car owner location, with other vehicles or OEMs, thereby compromising the owner's privacy. The privacy of the proposed method is inherited from the BC where each node is known by a changeable PK. Each overlay node uses a different PK to communicate with other overlay nodes to protect its privacy by

avoiding being tracked by an attacker. Privacy-related and sensitive information, e.g. vehicle location, can be stored in a local storage

The proposed architecture protects automotive services and applications by providing security at different levels: the BC is the key security provider in our framework as it provides access control, data integrity, and confidentiality (by utilizing encryption methods). The OBMs key lists and the LBM provide access control, and the data integrity is achieved using hashes in transactions. In addition, the LBMs control transactions that are sent to the smart building or the vehicle to protect the vehicle. In the overlay, cluster members monitor the OBM (i.e., CH) to detect if there is any misbehavior and if they have not received requested services properly. In such an instance, the overlay nodes would change the OBM within their cluster. The OBMs benefit from lists of keys to decide on the destination of a transaction thereby protecting cluster members from being attacked.

Selected attacks and security threats

In the following, we use selected security threats and attacks to evaluate the security aspect of the proposed architecture. Thereby, we focus on attacks affecting the security of a smart vehicle and define different attack scenarios allowing an attacker to take control of a vehicle:

Changing a software binary in the cloud: the attacker may seek to attack the cloud storage and in further consequence to manipulate the stored data to inject malware to a large number of vehicles. The signed hash of the stored data in the cloud storage as well as in the multisig transaction generated by the SW provider and the OEM, protect the vehicles against this attack.

Distributing a false update by claiming to be the OEM or SW update provider: the overlay nodes know the PK of the OEMs and the SW provider. Therefore, the attacker cannot claim to be either of these as it requires the private key associated with the PK of the relevant entities.

Distributed Denial of Service (DDOS) attack: In this attack, the attacker infects a large number of vehicles and uses them to launch a DDOS attack on a specific target, e.g. an OEM. In our architecture, vehicles are only accessible by those who have the vehicle's key in the overlay, and the vehicle owner should authorize requesters to access his vehicle using two key lists in the OBMs. In other words, unless a requester key is in the key lists of an OBM, the transaction is not considered valid. LBM increases the vehicle security as it authorizes all incoming and outgoing transactions and acts similar to a network firewall.

Linking attack: In this attack, an attacker tries to deanonymize a user by linking different pieces of data associated with the same anonymous user (i.e., linking the users' PKs). This would endanger the users' privacy. To protect against this attack, each user sets a different key for its data in the storage and for its overlay transactions.

Conclusion

In this paper, we propose a novel automotive security architecture based on BlockChain. Due to its distributed nature, the proposed architecture removes the demand for a central control system and allows novel automotive services.

The privacy of the involved users is ensured by changeable Public Key (PK) hiding the users' identity. The design security is increased as the OBMs provide access control for transactions, and the LBM controls the incoming and outgoing transactions to the vehicle. The architecture is able to support emerging automotive services by providing a secure and trustworthy way to exchange data while protecting the security of the end user.

We discussed several automotive use cases to illustrate the applicability of the proposed architecture. Additionally, we described possible attack scenarios and discuss how the proposed architecture is able to mitigate and inhibit these attacks.

References

[1] C. Valasek and C. Miller, Remote Exploitation of an Unaltered Passenger Vehicle, White Paper, 2015.

[2] Heinecke, Harald, et al., Automotive open system architecture-an industry-wide initiative to manage the complexity of emerging automotive E/E-Architectures, SAE Convergence, 2004.

[3] Steger, Marco, et al., SecUp: Secure and Efficient Wireless Software Updates for Vehicles, IEEE Euromicro Conference on Digital System Design (DSD), 2016.

[4] Steger, Marco, et al., Generic framework enabling secure and efficient automotive wireless SW updates, IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA), 2016

[5] W. Zeng, M. A. S. Khalid and S. Chowdhury, In-Vehicle Networks Outlook: Achievements and Challenges, IEEE Communications Surveys & Tutorials, 2016.

[6] Satoshi Nakamoto, Bitcoin: A peer-to-peer electronic cash system, 2008.

[7] Gavin Wood, "Ethereum: A secure decentralised generalised transaction ledger", *Ethereum Project Yellow Paper*, 2014.

[8] Yue, Xiao, et al. "Healthcare data gateways: found healthcare intelligence on blockchain with novel privacy risk control." *Journal of medical systems*, 2016

[9] F-secure, https://sense.f-secure.com/, Online accessed: 24-03-2017].

[10] A, Dorri, S.S. Kanhere, R.Jurdak, "Towards and optimized Blockchain for IoT", Second IEEE/ACM conference on Internet of things Design and Implementation (IoTDI2017), 2017

[11] A. Dorri, S. Kanhere, R. Jurdak., and P. Gauravaram, "Blockchain for IoT Security and Privacy: The Case Study of a Smart Home," In proceedings of the 2nd IEEE Workshop on security, privacy, and trust in the Internet of things (PERCOM), Hawaii, USA, March, 2017.

[12] Apostolos Kousaridas et al., "SYSTAS: Density-based algorithm for clusters discovery in wireless networks", IEEE 26th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC), 2015.

Biographies

Ali Dorri received his bachelor degree in Computer Engineering from Bojnourd University, IRAN, 2012. He then commenced his master degree in Computer Engineering in Islamic Azad University of Mashhad, IRAN, working on

Mobile Ad hoc Networks and security issues rising from this sort of network. He now is a Ph.D. candidate in University of New South Wales (UNSW), Sydney. His current research interest covers security and privacy concerns in the context of Internet of Things (IoT), Wireless Sensor Network (WSN) and Vehicular Ad hoc Network (VANET). Moreover, he is working on bitcoin and its applications on IoT.

Marco Steger is a senior researcher at the VIRTUAL VEHICLE research center in Graz, Austria. He received a masters degree from Graz University of Technology in 2013 with a thesis titled “Development and Evaluation of C2X applications” done in cooperation with BMW AG, Munich, Germany. His research interests encompass dependable wireless and automotive communication networks, security in wireless/mobile/automotive networks, dependable wireless sensor networks, automotive control units, automotive software development, advanced driver assistance systems (ADAS), and car-to-x communication and applications.

Salil S. Kanhere received his M.S. and Ph.D. degrees, both in Electrical Engineering from Drexel University, Philadelphia. He is currently an Associate Professor in the School of Computer Science and Engineering at the University of New South Wales in Sydney, Australia. His current research interests include Internet of Things, pervasive computing, crowdsourcing, embedded sensor networks, mobile networking, privacy and security. He has published over 150 peer-reviewed articles and delivered over 20 tutorials and keynote talks on these research topics. He is a contributing research staff at Data61, CSIRO and a faculty associate at Institute for Infocomm Research, Singapore. Salil regularly serves on the organising committee of a number of IEEE and ACM international conferences (e.g, IEEE PerCom, ACM MobiSys, ACM SenSys, ACM CoNext, IEEE WoWMoM, IEEE LCN, ACM MSWiM, IEEE DCOSS, IEEE SenseApp, ICDCN, ISSNIP). He currently serves as the Area Editor for Pervasive and Mobile Computing, Computer Communications, International Journal of Ad Hoc and Ubiquitous Computing and Mobile Information Systems. Salil is a Senior Member of both the IEEE and the ACM. He is a recipient of the Humboldt Research Fellowship in 2014.

Raja Jurdak is a Principal Research Scientist at CSIRO, where he leads the Distributed Sensing Systems Group. He has a PhD in Information and Computer Science at University of California, Irvine in 2005, an MS in Computer Networks and Distributed Computing from the Electrical and Computer Engineering Department at UCI (2001), and a BE in Computer and Communications Engineering from the American University of Beirut (2000). His current research interests focus on energy-efficiency and mobility in networks. He has over 100 peer-reviewed journal and conference publications, as well as a book published by Springer in 2007 titled Wireless Ad Hoc and Sensor Networks: A Cross-Layer Design Perspective. He regularly serves on the organizing and technical program committees of international conferences (DCOSS, RTSS, Sensapp, Percomm, EWSN, ICDCS). Dr. Jurdak is an Adjunct Professor at Macquarie University and James Cook University, and Adjunct Associate Professor at the University of Queensland and the University of New South Wales. He is a Senior Member of the IEEE.