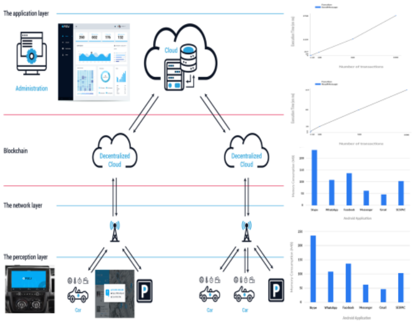
# Blockchain for the Internet of Vehicles: How to Use Blockchain to Secure Vehicle-to-Everything (V2X) Communication and Payment?

Rateb Jabbar, Noora Fetais, *Senior Member, IEEE*, Mohamed Kharbeche, *Member, IEEE*,
Moez Krichen, *Member, IEEE*, Kamel Barkaoui, and Mohammed Shinoy

***Abstract*—As the Internet of Things (IoT) is evolving, one of its rapidly developing components is the transformation of standard Vehicle Ad-hoc Networks (VANETs) into the Internet of Vehicles (IoV). Due to the exceptional progress in computation and communication technologies, the IoV has attracted the attention of researchers and commercial companies. Nevertheless, the primary issue regarding the IoV, and in particular to Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I), is establishing secure and instant payments and communications. To respond to this challenge, this work proposes a Blockchain-based solution for establishing secure payment and communication (PSEV) in order to study the use of Blockchain as middle-ware between different participants of intelligent transportation systems. The proposed framework employs Ethereum to develop a solution aimed at facilitating Vehicle-to-Everything (V2X) communications and parking payments. Moreover, the solution includes Android auto and application modules for automating the communication process. It was experimentally tested to assess its computational costs, communication expenses and the real-time aspect (RTA). The results of computational tests revealed that the developed solution is faster and more scalable than the existing solutions.**



***Index Terms*—Blockchain, smart parking, automated payments, automotive communication, Ethereum, internet of vehicles, Internet of Things, intelligent transport system, cloud and android.**

## I. Introduction

The purpose of the Internet of Things technology is to link various smart appliances and devices, facilitate their operations and data exchange. The smart devices, including wearables, smartphones, and sensors, obtain data to improve customer experience. In 2014, 13 % of businesses used IoT technology, and this number raised to 25 % with more potential increase in the future according to McKinsey [1]. Globally, 43 billion IoT-connected devices are expected by 2023 [1]. The revolution occurring on the IoV has been enabled by the IoT. Drawing upon Vehicular Ad-hoc Networks (VANETs), the IoV has begun its evolution with smartphones, which will eventually lead to a smart-car [2]. Conventional VANETs are aimed at improving traffic security and efficacy through enabling communication between vehicles in real-time due to advanced wireless access technology.

However, VANETs have not attracted commercial interest despite considerable potential concerning the security, efficacy of traffic and low costs of operation [3]. In particular, VANETs were not successfully commercialized because of the issues with ad-hoc network architecture [4] such as unstable Internet services [5], discordance with personal devices [6], unavailable cloud computing [7], insufficient precision of the services, and cooperative operational dependency of the network.

Currently, the increasing traffic causalities globally still remain a major problem despite the continual improvement of the road infrastructure and vehicle's modernization.

Manuscript received December 26, 2020; revised February 21, 2021; accepted February 21, 2021. Date of publication February 24, 2021; date of current version July 14, 2021. This publication was made possible by QUCP-CENG-2019-1 grant from the Qatar University. The statements made herein are solely the responsibility of the authors. The associate editor coordinating the review of this article and approving it for publication was Dr. Alireza Jolfaei. *(Corresponding author: Rateb Jabbar.)*

Rateb Jabbar is with the Department of Computer Science and Engineering, College of Engineering, Qatar University, Doha, Qatar, and also with the Cedric Laboratory, Computer Science Department, Conservatoire National des Arts et Metiers, 75003 Paris, France (e-mail: rateb.jabbar@qu.edu.qa).

Noora Fetais is with the Department of Computer Science and Engineering, College of Engineering, Qatar University, Doha, Qatar.

Mohamed Kharbeche and Mohammed Shinoy are with the Qatar Transportation and Traffic Safety Center, College of Engineering, Qatar University, Doha, Qatar.

Moez Krichen is with the ReDCAD Laboratory, University of Sfax, Sfax 3029, Tunisia.

Kamel Barkaoui is with the Cedric Laboratory, Computer Science Department, Conservatoire National des Arts et Metiers, 75003 Paris, France.

Digital Object Identifier 10.1109/JSEN.2021.3062219

Considering that the number of on-road vehicles is expected to continue its exceptional growth ensuring safety remains a paramount concern [8]. Accordingly, it is assumed that the rise in the number of vehicles will create space for a profitable market for "connected vehicles" despite the current challenges [9]. However, it is important to note that the concept of smart vehicles encounters significant challenges. The most critical challenge is to ensure the security of different participants in intelligent transportation systems. As a result of high connectivity, such systems are prone to malicious attacks. Furthermore, it is difficult to ensure privacy due to an exchange of sensitive data. To summarize, the following are the primary challenging issues:

- Centralization: Currently, centralized communication models represent the basis of smart vehicle architectures. Accordingly, the role of the central cloud server is identification, authentication, authorization, and connection of all the vehicles. Due to a low possibility of scaling this model, the failure of cloud servers is likely to compromise the whole network.
- Lack of Privacy: At the moment, communication architectures do not guarantee the protection of privacy. More precisely, the owner's permission is not required for data sharing related to the vehicle.
- Scalability: The extraordinary growth of embedded technologies has also significantly increased the use of miniaturized devices such as sensors and actuators. These devices tend to produce an enormous amount of data. Therefore, it is necessary to handle an ever-increasing number of devices and related data.
- Security: Attacks, including Man-In-The-Middle (MITM), are common in VANETs. Accordingly, it is difficult to guarantee their safety ( [10]–[12]). In February 2016 in Australia, Troy Hunt [13], certified as a Microsoft Security Professional, conducted an attack that allowed him to control vehicle properties of Nissan LEAFs from distance. In June 2016, a remote-control smartphone application was used to hack a Mitsubishi Outlander PHEV [14] by controlling non-driving critical elements of the vehicle systems (e.g., air conditioning, editing charger settings, and locking or unlocking doors). Moreover, Chinese Cybersecurity company Keen Security Lab has recently hacked Tesla [15] where researchers succeeded in managing parts of Model S Tesla (e.g., breaks and mirrors) from a distance of approximately 20km. From these illustrations, enhancing data transmission by developing more advanced solutions is crucial, including payment systems, to ensure efficiency, security, and integrity.

Blockchain is a cutting-edge technology that has recently gained considerable interest from academicians and business organizations. In particular, many articles and reports used Blockchain in several fields such as healthcare, cybersecurity, and transportation. Furthermore, Blockchain has increasingly gained prominence in Internet-of-Things (IoT) [16], [17]. Most importantly, Blockchain-based solutions can significantly improve the security of the IoT applications by establishing secure data sets exchange [18]. Indeed, Blockchain-based solutions aimed at ensuring privacy have been developed in the areas of personal data protection [19], protection against cyber-attacks [20], health care and other public services [21]–[23], electronic voting systems [24]–[26], databases operated by governments [27]. It is important to mention that personal information such as identity data and healthcare records [28], [29] are accessible with private keys ensuring total accountability and security. Recently, researchers have started using Blockchain technology to create secure, autonomous, and distributed intelligent transport solutions. Moreover, Blockchain ensures security of the IoV networks ( [30]–[32]). Nevertheless, the existing solution has limitations regarding supporting the real-time operation and Turing-Completeness aspect. To address this challenge of ensuring secure V2X communications and payments in intelligent transportation systems, this work proposes a Blockchain-based solution that employs Ethereum to create a smart payment system for parking (PSEV-Payment). Furthermore, the proposed solution is based on the assumption that owners of private parking spaces are inclined to share their parking lots whenever they are available. Moreover, this work introduces a Blockchain IoV Solution for Vehicles Communication (PSEV-Communication) that operates in real-time to guarantee the establishment of safe communication not only between the vehicles but between all participants in transportation systems. The main aim of this solution is to increase the utilization of parking spaces in urban areas, to reduce traffic congestion and to minimize the fuel consumption and time.

The main contributions of this paper are listed below:

1) A new IoT Solution (PSEV). PSEV contains three primary layers:
   - The perception layer includes Android Auto Application for Vehicles in charge of communications with vehicles, finding the parking space and managing the transaction between the car and the parking infrastructure, Android Application for Parking Space Renters and Android Application for the Parking IoT System.
   - The network layer that is in charge of sharing data between devices and the cloud relies on 3G/4G/5G and WiFi.
   - The application layer that contains a Blockchain solution is in charge of managing the parking payment system and enabling secure communication among all the participants in the systems.
2) The security, execution time, costs, immutability, integrity, confidentiality, and memory and power consumption of the developed solution were tested. The tests indicate that the system has exhibited high performance in all aspects as shown in Section V.

The rest of this paper is organized as follows. Section II presents an overview of the literature on the payments systems for parking spaces relying on IoT and the employment of Blockchain. Section III introduces the infrastructure and properties of the proposed system. The detailed description of nominal cases for utilizing the system are described in Section IV. Section V summarizes the test procedure as well as the results. In Section VI, the results of the computational tests was compared with the similar solution. Section VII concludes the paper and provides some directions for further research.

## II. Literature Review

### A. Blockchain for Vehicle-to-Everything (V2X)

In a technical sense, Blockchain [33] is a decentralized shared ledger based on chained, encrypted, and chronological blocks that store verifiable and synchronized data across the peer-to-peer (P2P) network. Therefore, Blockchain represents decentralized architecture and distributed computing paradigm. Its functions include manipulating data with self-executed program scripts (i.e., smart contracts), storing data with encrypted chained blocks, and creating data using distributed consensus algorithms. Furthermore, Blockchain is required for the functioning of cryptocurrencies such as Bitcoin [34]. All Blockchain system are based on the same consensus mechanism with slight differences. Fundamentally, each full-client node of the decentralized P2P network stores Blockchain data. Subsequently, the consensus mechanism is in charge of verifying and recording the data on the Blockchain network. In terms of advantages, Blockchain is publicly verifiable, fully distributed, transparent, decentralized and data cannot be modified. Many researchers have recently focused on the innovative Vehicle-to-everything (V2X), which integrates vehicles to establish smart communication between vehicles and different types of networks such as vehicle-to-vehicle, vehicle-to-human, vehicle-to-infrastructure, vehicle-to-road, and vehicle-to-sensor. Accordingly, Huang *et al.* [35] proposed an ecosystem model drawing upon Blockchain electric vehicle and charging pile management. In this ecosystem, Elliptic Curve Cryptography (ECC) is used to compute hash functions of charging piles of electric vehicles. In addition, Kang *et al.* [36] developed PETCON, a P2P electricity-trading system that illustrates localized and comprehensive operations of P2P electricity trading. This model is based on a consortium Blockchain method for analyzing, verifying, and sharing transaction records publicly. More importantly, it does not require a reliable authority. Moreover, Li *et al.* [37] developed CreditCoin that aimed at transmitting necessary announcements without disclosing users' identities. In the proposed solution, Blockchain is used via an aggregation protocol between vehicles to improve IoV information sharing confidence. Furthermore, Yang *et al.* [38] developed a Blockchain-based reputation system to evaluate data credibility in the IoV. The system assesses senders' reputation values to conclude if the message is true or false. Finally, the solution of Yong *et al.* [39] in the form of an Intelligent Transportation Systems (ITS) is used to solve security problems and performance limitations.

### B. Blockchain for Automotive Communication

The use of IoT devices is increasing in many field and has brought a lot of advantages, as well as, a plethora of network and security issues. A considerable portion of these IoT devices is vehicles, thereby firmly establishing the new term IoV. This sub-sector has witnessed many innovative concepts related to the development of smart communication between all the participants in the systems.

To solve numerous safety and security problems related to connected vehicles, many concepts focusing on Blockchain systems to secure communications and transactions have also been introduced. In this regard, Kang *et al.* [40] developed a management system focused on consortium distributed data relying on Blockchain used in VECONs. In this solution, Roadside Units (RSUs) verify and ensure secure data sharing data by using smart contracts. In addition, they collect data on the frequency of interactions, path resemblance, and timing of events to establish a secure data system. Nevertheless, there are drawbacks and security vulnerabilities related to the central architecture of traditional VANETs. To respond to these drawbacks, Leidin *et al.* [41] developed a framework for automatic organization, communication, and management for distributed vehicular networks based on Ethereum. Also, Reilly *et al.* [42] proposed a cutting-edge protocol for clients that is light and its communication integrity is implemented through IoT devices and smart cities. The authors considerably contributed to the field by introducing a mandatory authentication step of the origin of the data using an Ethereum address. Accordingly, trusted transactions are ensured. Moreover, the transaction of the developed protocol is less costly than when the Bitcoin-based NeuroMesh protocol [43] is used. It is worth to mention that this application can be employed in a limited way because of specific drawbacks. First, the solution requires an operating system, and consequently, it cannot be supported by all devices. Second, the solution is not adequate for applications that should operate in real-time due to the possibility that transactions are delayed.

Furthermore, vehicular automotive systems can be easily attacked during the updating process. To solve these safety challenges, Falco *et al.* [44] proposed an automotive mechanism to validate vehicular data sources and ensure data integrity, immutability, and reliability. The developed solution guarantees network scalability and employs limited computing and networking resources efficiently. A consensus voting system based on Distributed Hash Tables (DHTs) is implemented for validating changes in data such as distance travelled, car identification, and software updates. Moreover, secure data sharing is provided by employing Blockchain technology.

Additionally, secure inter-network communication of IVs is required to ensure trust and data reliability (provenance) and adequate and reliable VANET operation. A Blockchain-based protocol developed by Javaid *et al.* [45], [46] is based on smart contracts with a dynamic proof-of-work (dPoW) consensus algorithm, certificates, and Physical Unclonable Functions (PUF). This protocol manages a list of registered vehicles and, accordingly, distinguishes registered and malicious vehicles. Besides, the authors [47] aimed at improving the issue of authentication in the IoV, considering that security under strict latency is an essential requirement. Standard authentication protocols for vehicular ad-hoc networks are characterized by frequent authentication, which increases overhead while simultaneously reducing application traffic. However, in spite of this, vehicles remain a primary target of cloning, side-channel, and physical attacks due to their mobility. Thus, the purpose of this work is to introduce an efficient authentication protocol in the IoV based on PUF to ensure security by enhancing the throughput of application layer packets and reducing the overhead of authentication.

In addition, physical ultrasonic audio and visual light channels were used and proposed by Rowan *et al.* [48] to develop robust and secure V2V communication and data sharing. First, in the verification phase, the vehicles' ID and position should be detected to initiate communication. Following this phase, the corresponding ID is allowed to share data. The developed solution entails an innovative protocol to develop the key relying on a public key Blockchain infrastructure employed in inter-exchange of vehicle sessions. The above-mentioned physical channels are used in the protocol.

Lastly, Jabbar *et al.* [49] developed a decentralized framework on the basis of Blockchain technology (DISV). DISV ( [50]–[52]) represents a real-time application specification that provides secure communication between all participants in the transportation system. The developed solution is composed of three layers: the perception is the first, the network is the second, and the application layer is the third layer. The perception layer takes the form of an Android application including two sub-systems. The Vehicle Data Collection System (VDCS) is the sub-system with the main purpose of obtaining information regarding the journey and the vehicle ( [53], [54]). The second sub-system is called the driver drowsiness detection. It aims to detect the drowsiness of the driver by acquiring information about his or her behavior( [55], [56]). This solution lacks scalability as it can not be used by a large number of vehicles in the same region.

To summarize, the majority of the existing published studies employ Bitcoin [34] as an implementation platform for Blockchain. Most importantly, as Bitcoin is essentially a cryptocurrency for selling and buying goods in a secure and anonymous way, therefore, smart contracts are not supported. Moreover, it does not possess the programming characteristics required for resolving computational problems aimed at allowing the transfer of various sensitive data. It is necessary to develop a Blockchain platform that is able to support Turing-Complete operations to consequently propose cutting-edge vehicle communication applications, which is the case with Ethereum [57]. Therefore, this work proposes an innovative vehicle communication and payment framework using Blockchain where the real-time aspect is based on the Ethereum platform, and accordingly, supports Turing-Completeness. The performance test has confirmed the real-time aspect of the developed solution.

### C. Blockchain for Payments

One of the first and foremost uses of Blockchain technology was to facilitate payments. Nowadays, there are approximately 7,000 cryptocurrencies in the market [58]. These currencies can be used to carry out payments for different services. The adoption of these services in the IoV is not very common. Researchers have introduced different approaches to IoV to maintain privacy during payments. Nevertheless, a lot of them addresses only the privacy exposure while the payment is being executed. Their objective is to hide identity, position, and time during the exchange of electric vehicles data and charging stations. In this paper, our work along with the above-mentioned advantages sets up a system that does automated detection and payments using Blockchain.

Both industry and academic community have focused on building Payment Channel Networks (PCNs). The first aspect aims at developing a PCN for intra-Blockchain operations, such as the Lightning Network [59]. The Lightning Networks transfers Bitcoin between parties through an off-chain link instantly and without a transaction fee. A similar concept was applied by Raiden [60] to develop a PCN for Ethereum. The second aspect aims at developing inter-Blockchain operations for transfers between various cryptocurrencies that do not require costly confirmation. Notable cases are Inter-Ledger [61] and Atomic-CrossChain [62]. Currently, the PCNs are in the initial phase of development. Therefore, privacy in the routing of the payment has not been achieved yet. The researchers have started to cooperate with the Blockchain community to ensure privacy in PCNs ( [63], [64]). All aforementioned studies assumed the availability of a perfect decentralized P2P PCN topology and developed their privacy-protecting routing solutions using this assumption. It is important to mention that research on the issues related to network formation and the impact on privacy are still lacking. Furthermore, in general, the researchers assume that individual pairs can enable other users to employ their channels by offering rewards such as rewarding fees.

### D. Research on Parking Payments

This paper primarily focuses on the development of the system capable of conducting automatic payments for parking spaces without compromising safety and security. To the best of our knowledge, few studies have been conducted, particularly regarding the collection of parking and payments data and related-information exchange.

Lu *et al.* [65] proposed SPARK which is a secure navigation scheme aimed at big parking lots to ensure convenient parking services. The authors included specific protections such as parking navigation in real-time and protection against theft.

Zhu *et al.* [66] worked on vehicular networks to include a smart parking system that is anonymous. This solution generated a trusted environment by short randomizable group signatures. According to the assessment results, this system ensures security and privacy at a low cost owing to the low computation requirements. Also, Rutgers University proposed a ParkNet system [67] aimed at assessing a number of available street parking places using an ultrasonic range facing the passenger side and vehicles with GPS to identify available parking lots. This system includes a central server whose role is to collect and analyze the data and subsequently transfer them to the drivers in real-time.

Ni *et al.* [68] proposed a parking navigation system (CPARN) based on the cloud aimed at protecting privacy by ensuring communication between the vehicles. Security and privacy are ensured while the server informs the drivers about the available parking spaces.

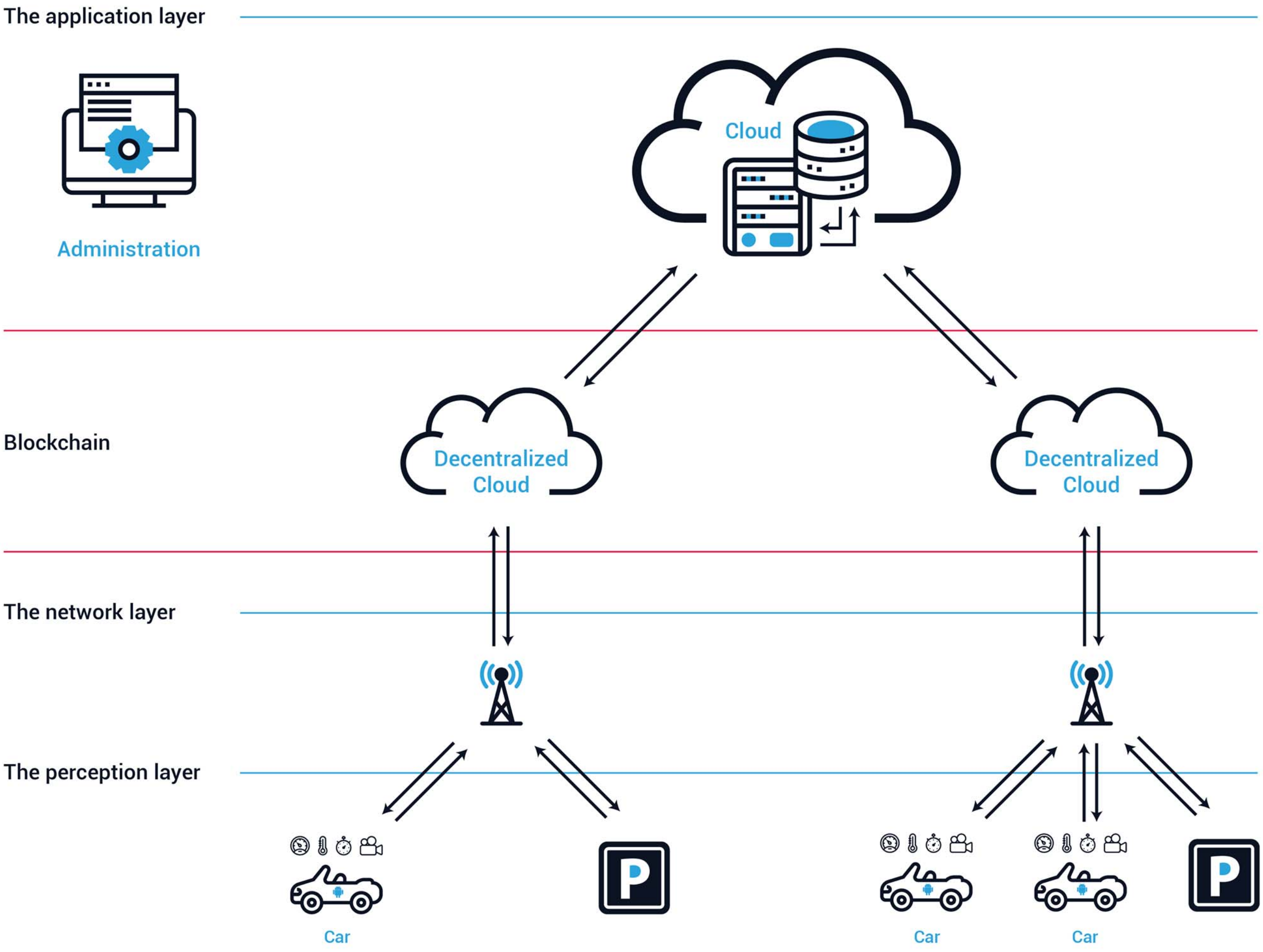


Fig. 1. The developed Internet of Vehicle architecture PSEV.

Parq [69] proposed a solution that uses tokens to pay for the parking spaces, whereas transactions are secured using Blockchain contracts. Similarly, ParkRes [70] shows the empty parking spots for the drivers on a map that can be booked in advance and paid through the Blockchain network.

As described, in all these references, the drivers use smart phone applications. These apps neither support direct integration with the related vehicle environment nor automated payments. Thus, our proposed solution represents a considerable improvement of the existing solutions as it uses Blockchain to ensure secure access to available parking spaces, navigation, and automatic payment procedure. Moreover, this solution includes parking lots owned by private as well as public buildings, which is elaborated in more detail in the following section.

## III. Proposed Architecture

### A. System Overview

This paper introduces IoV solution with Real-Time Application (RTA) aimed at establishing secure communications and payments in the transportation systems. The proposed solution uses IoT architecture and consists of the perception, network, and application layers. The layers are shown in Table I below.

The proposed architecture as shown in Fig. 1 will be explained in more detail in the following section.

### B. The Perception Layer

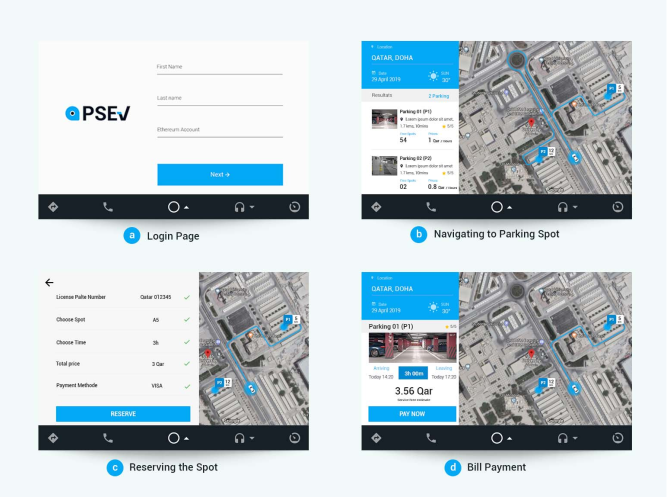
This layer includes three separate applications for vehicles and parking infrastructure required to ensure secure communication and payment. It consists of an Android Auto Application for Vehicles, Android Application for Parking Space Renters, and Android Application for the Parking IoT System. Using these three applications, the users can a) establish communication b) manage their own parking space, and c) conduct payments.

*1) Android Auto Application for Vehicles:* This is the car interface employed by the user for sending and receiving messages and searching, navigating, and paying for parking. In addition, this application detects the proximity of the car to the parking payment station to initiate the Blockchain-based transaction. The content of the Android application is depicted in Fig. 2 and Fig. 3.

TABLE I
THE MAIN FEATURES OF THE DEVELOPED PSEV SOLUTION

| Layers | Developed Solution | Main Features |
|---|---|---|
| Perception layer | Android Auto Application for Vehicles | This auto-based application is the interface in the car to be employed by the user. This console is used to send and receive messages, and search, navigate, and conduct parking payment when parking is required by the driver. |
| | Android Application for Parking Space Renters | This Android application allows the owners of building/parking space to manage and set specific time slots in which their parking spaces are available for rent to the public. |
| | Android Application for the Parking IoT System | This is an embedded application in the IoT parking payment machine that detects nearby cars and grants access to parking lots after submitting a request. |
| Network layer | | Connects the application in the perception layer to other network devices, servers, and smart things. |
| Application layer | Blockchain Application | Manages payment and establishes communication between vehicles and other actors in the transportation system. |
| | Central Cloud Server | The central cloud server serves as a host of the management and administrative software. |

Fig. 2. Main navigation pages of the Android Auto Application for Vehicles.



License Palte Number Qatar 012345 ✓
Choose Spot A5 ✓
Choose Time 3h ✓
Total price 3 Qar ✓
Payment Methode VISA ✓
RESERVE

Fig. 3. Screenshot of the primary navigation pages of the Android application.

As it is shown, the interface contains the login page that appears when the application is turned on. Following the authentication, the screen shows the map with the nearby parking locations. Once the user selects the appropriate location, the application will provide the navigation information. As the user reaches the prescribed parking area, the application shows the list of parking slots available at the selected location and their time slots. The user selects the most suitable slot according to their needs. After selecting the slot, the application initiates the timer.

When leaving the parking, the driver should click "STOP" on the timer and the application shows the summary of the transaction as well as the parking fee to be paid. The user can also access the history tab to check all previous transactions and their details.

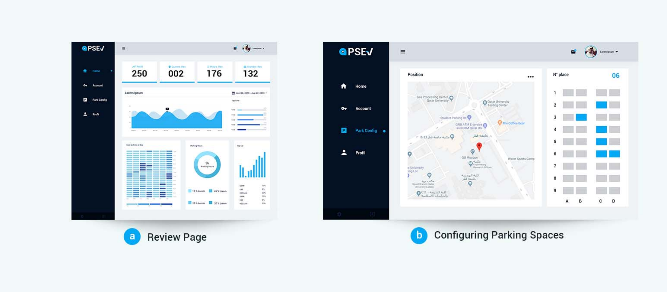
*2) Android Application for Parking Space Renters:* This Android application enables the owners of building/parking space to manage and set time slots in which their parking spaces are available for rent to the public. Hence, the parking owners can generate revenues from the unused parking slots.

This application has mainly three pages. The first one is the login page for authentication. The second is a parking slot configuration page that allows the user to define the time slots in which his parking space is available for rent. The renters can access the lists of previous transactions, if any, and the total amount earned for the given parking space. Finally, a settings page serves for account management and administrative tasks.

*3) Android Application for the Parking IoT System:* This is an embedded application in the IoT parking payment machine that will detect the nearby cars and grant access to parking slots after submitting the request by the drivers. This payment portal manages the authentication and time slot management as well as the payment transaction between the car and the parking space provider. This application plays the role of the parking access machine that guarantees physical access to the parking.

## C. The Network Layer

The purpose of the network layer is to establish the connection between parking payment portals located in the perception layer (Android Auto Application for Vehicles, Android Application for Parking Space Renters, and Android Application for the Parking IoT system), smart cars, and the central cloud server in the Blockchain sub-layer. This layer



a Review Page

b Configuring Parking Spaces

Fig. 4. Screenshot of the developed web application.

transmits and processes the IoT data securely through WiFi or mobile internet (3G/3G+/4G/5G).

## D. The Application Layer

This layer hosts the server and applications responsible for the communication and payment systems. It meets the particular needs of the management system of the devices. The central cloud server and the communication system are the main components of this layer. They are operated using the Blockchain network.

*1) Central Cloud Server:* The management and administrative applications are hosted in the central cloud server. The cloud monitors the systems and functioning of the infrastructure as administrator. Thus, it has the function of hosting and administrating services of all payment portals and cars involved in the system. The user employs the Android Auto Application for Vehicles to purchase the ticket. Next, the central server notifies the IoT devices in the same area to communicate with each other via one of the existing Blockchain cloud instances. More precisely, IoT devices receive an invitation to communicate via the Blockchain sub-layer.

The web application as shown in Fig. 4 displays the analysis, operations conducted as well as the configuration of the parking slots.

It serves as a tool in the application layer employed to establish interaction among various elements of the IoT solution, including the administrative web site, a database server of the devices that interact with the Blockchain, and the embedded systems at the payment portals. Windows Azure cloud service with REST API calls has been utilized to host this service. Several scientific tools that can analyze the traffic flow at different parking spots are also included in this web application in order to investigate and understand the parking demand requirements at different locations of the city. Hence, this serves as a central administrative platform to be used by the administrator to troubleshoot the system if necessary and ensure smooth functioning.

*2) Blockchain Sub-Layer:* The Blockchain is responsible for facilitating communication among various elements and road users of smart cities and the payments between cars and parking providers. More precisely, every time a user intends to conduct a parking payment, this transaction is broadcasted to the Blockchain sub-layer and validated. Completed transactions become a part of the block, which makes them valid and immutable. The Blockchain sub-layer and the Android application are merged to generate decentralized apps (Dapp,dApp, or DApp). Decentralized application operate on a decentralized P2P network such as Blockchain. The Android application is the front-end, while the Blockchain sub-layer is the back-end parts of this decentralized application.

The smart contract functions are deployed in all Ethereum nodes, and the mobile application uses the Blockchain for sending a message. The mobile and node-endpoints establish the communication. This solution employs exceptionally reliable Web3.Js for the Android framework.

The used Blockchain sub-layer contains the following systems:

## E. Access Management System

The Access Management System enables the identification and authentication of different participants of intelligent transportation systems to exchange data. The following modules are located in the Blockchain sub-layer in the Access Management System:

- Vehicle Management Module: The Vehicle Management Contract is an important element of this module. It enables adding, modifying, and deleting a vehicle.
- Parking Management Module: The Parking Management contract is also an important element of this module. It enables adding, modifying, and removing a parking lot in the system.
- Participant Management Module: The Participant Management Contract enables adding, modifying, and suppressing another participant that can send messages to the system, including roadside electronic signs, traffic lights, and radars.

## F. Parking Payment Management System

This system allows the management of the parking. It contains a smart contract that enables the providers to sell tickets and get numbers for the available slots. In addition, it contains a smart contract that enables clients or vehicles to purchase the available tickets. As outlined previously, these modules are located in the Blockchain sub-layer in the Parking Payment Management System.

## G. Communication Management System

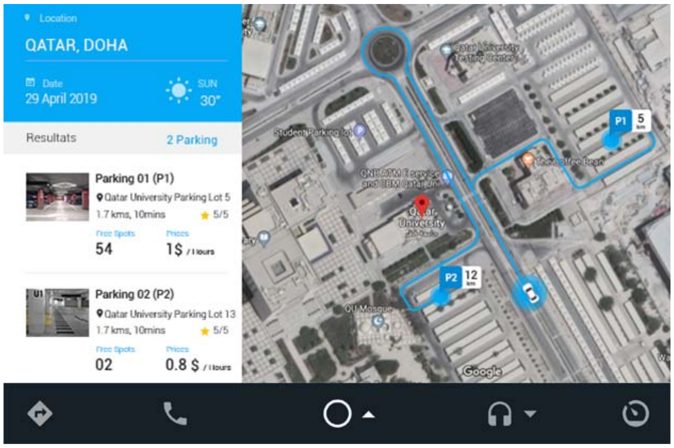
The smart contract of the Communication Management System has two primary functions (SendMessage and ReadMessage). SendMessage function transmits a new message via the Blockchain network. This action is based on the ETH amount where the sender pays per unit of gas for mining the message. On the other side, ReadMessage function allows the participant to access the data through a device that is a part of the Blockchain network.

# IV. Nominal Scenarios

## A. Parking Management System

In this section, two nominal cases for utilizing the system are shown below:

1) Requesting parking
2) Providing parking as a supplier



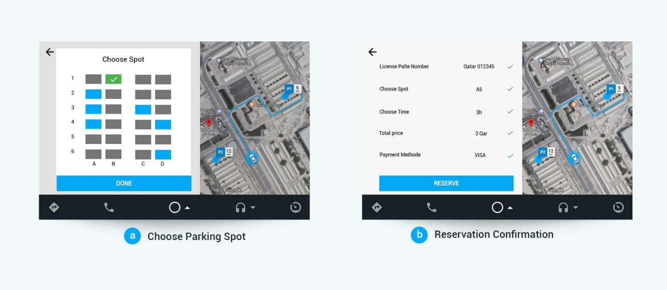
Location
QATAR, DOHA
Date
29 April 2019
SUN 30°

Resultats 2 Parking

Parking 01 (P1)
Qatar University Parking Lot 5
1.7 kms, 10mins ★ 5/5
Free Spots 54 Prices 1$ / I hours

Parking 02 (P2)
Qatar University Parking Lot 13
1.7 kms, 10mins ★ 5/5
Free Spots 02 Prices 0.8 $ / I hours

Fig. 5. Screenshot of Android Application for Vehicles displaying requesting for parking.



Choose Spot

| | A | B | C | D |
|---|---|---|---|---|
| 1 | | ✓ | | |
| 2 | | | | |
| 3 | | | | |
| 4 | | | | |
| 5 | | | | |
| 6 | | | | |

DONE

a Choose Parking Spot

| | | |
|---|---|---|
| License Palte Number | Qatar 012345 | ✓ |
| Choose Spot | A5 | ✓ |
| Choose Time | 3h | ✓ |
| Total price | 3 Qar | ✓ |
| Payment Methode | VISA | ✓ |

RESERVE

b Reservation Confirmation

Fig. 6. Screenshot of the list of prices and the available slots.

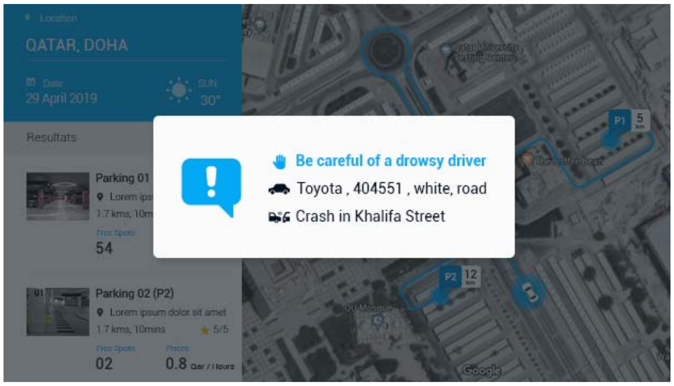
*1) Requesting for Parking:* The procedures involved in the transaction are listed as follows: the driver taps on the “Find Parking” icon When looking for a parking. The application sends this request to the central server and the server returns the list of different locations nearby that are available with their respective prices. The user clicks the preferable available parking slot. The application turns on the navigation to the selected destination, as shown in Fig. 5.

Once at the destination, the payment portal will authenticate the car using the Blockchain protocols and verify with the server to make sure the customer is enrolled in the Blockchain and is a valid customer. Next, the user is provided with the parking spaces inside the parking, with the list of prices and the available time slots. The user selects the preferred one and the reservation is confirmed as shown in Fig. 6.

The payment portal, which is the control unit that grants access to the physical location, opens the gate and allows the car to enter. The user looks for the selected parking space and parks his car at the designated location. Once parked, the application starts the timer, displays the time and shows the fees incurred on the screen.

Once the user intends to leave the parking lot, he drives out of the parking lot, and the payment portal notices that the car is exiting. The application then displays the total time and cost incurred. The Android Auto Application for Vehicles will pay the parking lot machine automatically for the above-said amount over the Blockchain. This transaction is then verified and noted in the Blockchain channel.

*2) Providing Parking as a Supplier:* The nominal procedure involved in a user renting out his parking space is as follows:



Be careful of a drowsy driver
Toyota , 404551 , white, road
Crash in Khalifa Street

Fig. 7. Screenshot of a message received to alert the driver.

The user logs into the Android Auto Application for Vehicles via the login page. The user selects the parking lot from his list, sets the time slots appropriately as per his requirements, and submits the request for broadcasting to the Blockchain. The server receives that information and makes those parking spots available for the set duration to all drivers using the service.

## B. Communication Management System

We used the following scenario: sending alert messages in case of drowsiness to develop and test the prototype of PSEV-Communication. In the case of a drowsy driver, the cars nearby are alerted by a message transmitted through the Blockchain. We included the factor of driver drowsiness, considering that it is one of the leading causes of motor vehicle crashes that was confirmed by a study [71] conducted by the AAA Foundation for Traffic Safety. Essentially, this report implied that over 5,000 Americans lost their lives as a result of sleep-related vehicular crashes. The car sends to the Blockchain sub-layer a message that contains essential information, such as who is the sender, time, content of message, and the position. After the mining process is completed, the message becomes a part of the smart contract, and accordingly, any device that is connected to this server Blockchain sub-layer is able to access it as illustrated in Fig. 7. The central server receives a copy of the message in the Blockchain.

# V. System Analysis and Performance

Different methods can be employed to evaluate the performance of the software solution ( [72], [73]). The most critical part of this system is the assessment of the particular features needed for the proper functioning of the solution. The main features of the proposed solution are integrity, cost, consistency, confidentiality, execution time, immutability, and memory and power consumption. The smooth operation of the solution requires that all these features achieve high performance. Therefore, this paper evaluates specific features to evaluate the general performance of the proposed solution.

## A. Security

Some testing must be considered to ensure a secure transaction using Blockchain technology. OWAPS guidelines for mobile and web applications, Mythx, and SCSVS are applied to overcome challenges and to ensure the security of the Blockchain-based solution. Importantly, the testing depends on the methods of implementation in the applications. These factors provide adequate proper validation and verification approach for the Blockchain-based applications.

*1) Mythril and MythX:* We conducted security tests to ensure the resiliency of the system against malicious attacks. The first test was performed to detect security vulnerabilities in our smart contracts, which represent the core of our system. In this test, we scanned the smart contracts against possible vulnerabilities using the security tool named MythX [74] a Smart contract security service for Ethereum. MythX is divided into three tools: Maru for static analysis of smart contracts, Harvey for smart contract fuzzing, and Mythril for EVM bytecode symbolic execution. This service is aimed at analyzing the security of smart contracts at compiled bytecode and source code levels. This test did not reveal any vulnerabilities in the smart contract listed in the MythX Smart Contract Weakness Classification (SWC) Registry. For instance, the MythX report did not register assert violation, unprotected Ether withdrawal, or integer overflow/underflow. However, although smart contracts do not contain vulnerabilities as they are based on the blockchain, it should be underlined that the system is prone to 51 percent attack.

*2) OWASP:* OWASP is a non-profit organization that provides unbiased, practical guidelines to developers and security professionals by identifying primarily crucial web apps vulnerabilities. The Open Web Application Security Project (OWASP) Foundation's [75] top web vulnerabilities were used for the assessment of the security of the central server of the PSEV system, web services, and websites.

Table II shows the ten most common attacks identified by the vulnerability assessment analysis and necessary security requirements that should be incorporated into the system to reduce vulnerability combat attacks. However, it is noteworthy to mention that the attackers continually perform increasingly complex attacks. Therefore, it is not possible to completely eliminate the attacks. Instead, the prevention aims to minimize the damage and decrease chances for hacking the system.

Table III outlines specific vulnerabilities of Ethereum's architecture and technology regarding smart contracts. The most common attacks on Ethereum are related to the external call; however, the attackers also use a loop in the smart contract by exploiting specific functions. The proposed solution is based on private Blockchain, and accordingly, it is of high security. In addition, Table III presents several recommendations.

*3) SCSVS:* Smart Contract Security Verification Standard (v1.1) [76] represents a list aimed at standardizing the smart contract security for vendors, security reviewers, architects, and developers. The purpose of this checklist is to minimize vulnerabilities and security problems. Therefore, it offers guidance for all stages of the smart contract development cycle, starting with design and ending with implementation. The primary objective is to establish a high quality code of the smart contracts. Accordingly, it is required to identify and mitigate vulnerabilities. Furthermore, the objective of the Smart Contract Security Verification Standard (v1.1) is to offer a straightforward and reliable assessment of the security of smart contracts regarding the percentage of SCSVS coverage and a checklist for security reviewers.

In this work, the recommendation of the Smart Contract Security Verification Standard was respected in the following Area: In the domain of architecture, design, and threat modelling, we tested the verification of a component that uses events to supervise the contract activity and of a policy for tracking new security bugs and updating the libraries to the newest secure version. Furthermore, we respected the consistency of the business logic in the contracts. Either none of the contracts should be allowed to conduct changes or all the contracts should be allowed. In addition, we applied the verification using code analysis tools to identify potentially malicious code. Moreover, we applied the newest version of Solidity and respected that the controlled, minimal acceptable value of cryptocurrencies is maintained on the contract.

In the domain of access control, the presence of the principle of least privilege tested: other contracts must have particular authorization to access data or functions. Furthermore, we respected that calling external contracts is possible in urgent cases only. In addition, we respected the basis of the contract on the data given by the right sender. Thus, the contract must not be based on tx.origin value. Also, access controls must maintain all user and data attributes in the trusted contract. Accordingly, the manipulation by other contracts is prevented if they do not have particular authorization.

In the domain of Blockchain data, we tested if data stored in the contracts is not private or safe (private variables are also included). It is also necessary to respect and verify that the contract does not employ literals as mapping keys for mappings, but that global constraints are employed for the prevention of the Homoglyph attack. Finally, we applied the identification of the list of sensitive data that the smart contract processed. In the domain of communications, we tested if the libraries were identified. The smart contract depends on the libraries for operation, but they are not a component of the app. We also respected that the contract does not employ hardcoded addresses, except in necessary cases. In the case of using a hardcoded address, the contract must be audited. In addition, we applied a centralized implementation of libraries and contracts that call external security services, that untrusted contracts and delegatecall are not used together, and that the result of low-level function calls such as call, delegatecall, and send are inspected and included as well. In the domain of gas usage limitations, we tested the anticipation, definition, and distinctive limitations that cannot be exceeded of the usage of gas in the smart contract. Therefore, gas exhaustion is not caused by malicious input and code structure. In this domain, we also respected that two kinds of the addresses are taking into account when employing the send function. It is more expensive to send to the contract address than to the personal address. In addition, we applied the principle that the iteration of the contract over unbound loops does not occur, the contract does not inspect if the

TABLE II
OPEN WEB APPLICATION SECURITY PROJECT (OWASP) TOP WEB VULNERABILITIES AND SECURITY AND PRIVACY REQUIREMENTS

| No. | System Attack | Description | Security and Privacy Requirements |
|---|---|---|---|
| A1 | Injection | CRLF, LDAP, and SQL are injection attacks performed by sending an untrusted data to an interpreter. Consequently, the command is executed without proper authentication. | The system prevents such attacks by ensuring the separation between the data and queries and commands. |
| A2 | Broken Authentication | The attacker uses Broken Authentication and Session Management vulnerabilities due to accessing the passwords' data base or the session ID. Broken Authentication can be defined as manipulation of authentication mechanisms. | Broken Authentication is prevented by selecting strong passwords and ensuring adequate storage mechanisms. |
| A3 | Sensitive Data Exposure | The attackers can exploit Applications and APIs without protection of sensitive data, such as financial data, passwords, and usernames. As a result, the attackers can obtain and misuse data by for example committing fraud and stealing identities. | The system must entail SSL. AES-256 and similar encryptions should be used to transfer the data. All insecure obfuscation techniques must be detected. |
| A4 | XML External Entities (XEE) | XL is uploaded by the attacker and hostile content is inserted in an ML document. Consequently, the attackers can manipulate vulnerable integrations, dependencies, and code and misuse vulnerable XML processors. | The system must entail the REST paradigm. The sensitive data serialization should be minimized. It is recommended to use JSON and similar formats. |
| A5 | Broken Access Control | The attacker misuses specific functionalities of the application to exploit Broken Access Control. In addition, the attacker can modify an URL further to reach even more functionalities. | The system must entail a strong access control mechanism. |
| A6 | Security Misconfiguration | If the web server fails to protect directories and files due to outdated and insecure configurations, the attacker can perform manipulation. | It is necessary to follow best practices for hardening properly all application components, including the operating system, language runtime, and server. |
| A7 | Cross-Site Scripting | The victim's browser or service endpoint can be used to perform the script. The purpose of performing the script is to hijack the session or to redirect the user to malicious websites. | When adequate data validation is applied, the malicious data cannot attack the database or the website. |
| A8 | Insecure Deserialization | The attackers use insecure deserialization flaws to delete or modify serialized objects written on the disk), perform the remote code in the application, upgrade privileges, or carry out injections. | The system must entail SSL. |
| A9 | Using Components with Known Vulnerabilities | The full privilege is needed to run vulnerable components (e.g., frameworks and libraries) | The system should use exclusively approved enterprise libraries. |
| A10 | Insufficient Logging and Monitoring | Breaches can be identified months after they were conducted if there are inadequate logging and ineffective integration and security incident response systems. Consequently, the attacker can access other systems and establish a persistent threat. | The system must entail monitoring systems such as Appdynamics and Dynatrace to ensure that proactive alerts are sent by following predefined rules. |

address is a contract via extcodesize opcode, and that the contract does not create trivially pseudorandom numbers using the information extracted from Blockchain (e.g., seeding with the block number).

In the domain of business logic, we tested if the business logic flows of smart contracts follow a sequential step order that is pre-designed. Accordingly, none of the steps can be skipped. We also respected that the contract has business limits that are appropriately enforced, the contract logic is not based on the balance of the contract (e.g., balance == 0), and the contract does not send funds automatically and ensuring that users perform separate transactions to withdraw funds. In addition, we applied that the inherited contracts do not include identical functions and careful specification of the order of inheritance.

## B. Cost

In standard Blockchain deployment, cryptocurrencies are incentives employed to reward network nodes/operators, which ensure consensus and data integrity, and accordingly, ensure the decentralized ecosystem. However, considering costs required for storing information and performing computations, Blockchain users must pay for incentives. Therefore, this selection examines the costs related to the services run by a DApp with these employments. Furthermore, the study compares those expenses with the costs incurred in widely used centralized and proprietary systems. In particular, it is necessary to perform cost estimation when a DApp provides services to numerous providers and clients. Moreover, the assessment of the cost-effectiveness is critical when Blockchain is used to replace conventional elements of the parking systems to improve interoperability. TESTNET Ropsten of the Ethereum network deployed the prototype of the smart contract and accordingly, assessed how cost-effective is the solution. This part of the paper evaluates the expanses needed to create and execute the smart contract. More precisely, the Blockchain sub-layer contains the modules, in which smart contracts are located. The following rates, valid in May 2020 were used: 1 gas = 1 wei (0.000000001 ETH) and 1 ETH 209 US. The lowest gas value that can be used in a transaction is 1 wei; the average gas value was approximately 0.006252 Ethereum

TABLE III

ETHEREUM TOP WEB VULNERABILITIES AND SECURITY AND PRIVACY REQUIREMENTS

| No. | System Attack | Description | Security and Privacy Requirements |
|---|---|---|---|
| AE1 | Re-Entrancy (DAO attack) | The smart contract of Ethereum is able to call and use codes of other external contacts. However, these contacts could be hijacked and consequently forced to perform new codes by a fallback function. | The transfer function must be prevented from transmitting more than 2,300 gas with the external call. As a result, the destination address/contract is not able to call another contract. |
| AE2 | Arithmetic Over/Under Flows | Fixed-size data types for integers are determined by the Ethereum Virtual Machine (EVM). In the case of an unchecked user input, attackers are able to use variables and carry out calculations in numbers outside the range of the data type in which they are stored. | Mathematical libraries can be designed or used to replace multiplication, subtraction, and addition, which are standard math operations, to ensure protection against under/overflow vulnerabilities. |
| AE3 | Delegatecall | Ethereum developers employ the DELEGATECALL and CALL opcodes for modularizing the code. Notably, DELEGATECALL may lead to unintended code execution. | Solidity possesses the library keyword for the implementation of the library contracts (Seethe Solidity Docs) As a result, the library contract is stateless and non-self-destructible. |
| AE4 | Default Visibilities | Solidity functions entail visibility specifiers for determining in which cases functions can be called. If specifiers are used improperly, the attackers can exploit resulting vulnerabilities. | The contract must specify the visibility of all functions. |
| AE5 | Short Address/Parameter Attack | The ABI specification encodes the parameters passed to the smart contract. It is possible to send encoded parameters shorter in length than the expected ones. / | The system should validate the input before it is sent to the Blockchain. |
| AE6 | Unchecked CALL Return Values | There are several methods for carrying out external calls in solidity. Most commonly, ETH is sent to external accounts by using the transfer method. In the case of versatile external calls, the send () function is used. In solidity, the CALL is used directly. | In the case of the external transaction reverts, it is recommended to use the transfer() function instead of send() transfer() reverts whenever it is possible. |
| AE7 | Denial Of Service (DOS) | A DDoS attack on Ethereum Blockchain implies the attacker employing resources of the network to prevent minors from recording or handling other transactions. | Contracts should be prevented from looping through data structures that permit artificial manipulation by external users. |
| AE8 | Tx.Origin Authentication | Phishing attacks can exploit contracts in which tx.origin variable is used for user authorization. In these attacks, users are manipulated to perform authenticated actions on the vulnerable contract. | tx.origin should not be used for authorization in smart contracts. |

TABLE IV

FUNCTIONS COST OF THE SMART CONTRACT BASED ON THE USED RATES

| System | Function | Gas Used | Price (US ˜Dollars) |
|---|---|---|---|
| | Deploy System | 0,14437926 ETH | $30,70 |
| Access Management | AddVehicles | 132797 | $0.08446 |
| | GetVehiclesByAddress | 0 | 0 |
| | GetAllVehicles | 0 | 0 |
| | AddParking | 150346 | $0.09561 |
| | GetParkingByAddress | 0 | 0 |
| | GetAllParking | 0 | 0 |
| Communication Management | SendMessage | 89280 | $0.05677 |
| | ReadMessage | 0 | 0 |
| Parking Payment Management | SellTicket | 166854 | $0.10613 |
| | GetNumberOfTickets | 0 | 0 |
| | GetTicketsForSale | 0 | 0 |
| | BuyTicket | 43078 | $0.2739 |

at the time of evaluation.

$$1 \text{ Gas} = 0.006252 \text{ Ethereum (ETH)}$$
$$\text{Gas Price} = 4{,}652{,}309.7157 \text{ Gwei}$$

Table IV gives an overview of the execution costs of the functions that are most commonly called. As summarized, the highest cost of $30.7 US is incurred to generate and deploy the prototype on the Blockchain. However, it is worth mentioning that only one payment is required to establish and deploy the system. Moreover, if Truffle [77] is removed, this cost decreases. Truffle is the framework used to build, deploy, and manage smart contracts. It possesses a Migrations contract for managing the cycle of deployment. Our analysis revealed that “AddVehicles”, “SendMessage”, “BuyTicket”, and “SellTicket” are the most frequently called functions. The “BuyTicket” costs on average $0.028 US. Furthermore, mining is not required to get messages from the blocks when calling the functions “GetVehiclesByAddress”, “GetAllVehicles”, “GetParkingByAddress”, “ReadMessage”, and “GetTicketsForSale”. Consequently, updates are not needed for the smart contract, and these functions do not incur further costs.

## C. Execution Time

The smart contract prototype was deployed via the Ethereum network TESTNET Ropsten. Table V shows clearly that the server’s response time of the functions “GetTicketsForSale”, “ReadMessage”, “GetParkingByAddress”, “GetAllVehicles”, and “GetVehiclesByAddress” are significantly shorter than the response times of other functions as mining is not required for their interaction with the smart contract. Nevertheless, for calling any function, it takes a minimum of 2 and a maximum of 57 milliseconds. Therefore, PSEV has been proven to be a real-time application.

The execution time is a fundamental criterion for the assessment of communication systems such as

TABLE V
EXECUTION TIME OF MOST CALLED FUNCTIONS OF THE SMART CONTRACT IN MILLISECONDS

| System | Function | Time (ms) |
|---|---|---|
| Access Management | AddVehicles | 28 |
| | GetVehiclesByAddress | 7 |
| | GetAllVehicles | 11 |
| | AddParking | 32 |
| | GetParkingByAddress | 9 |
| | GetAllParking | 15 |
| Communication Management | SendMessage | 4 |
| | ReadMessage | 2 |
| Parking Payment Management | SellTicket | 43 |
| | GetNumberOfTickets | 18 |
| | GetTicketsForSale | 12 |
| | BuyTicket | 57 |

PSEV-Communication. It is worth to mention that minor delays in sending or receiving messages might occur and would result in serious disruptions of the system. Adding large amount of messages at the same time to the smart contract, is essential to ensure the adequate operation of the framework for communication among all participants.

Fig. 8 demonstrates that the "ReadMessage" function takes significantly less time to respond than the "SendMessage".

More precisely, it takes between 2 milliseconds and up to 157 milliseconds when calling 1,000 requests of "ReadMessage" function. Accordingly, it does not make a significant difference in execution time if this function is called once or even 100 times. In contrast, calling the "SendMessage" function requires much more time as it is necessary to mine the message in order to add it into the smart contract. In fact, it takes only 19 milliseconds when ten requests of "SendMessage" function are received by the server and more than 2.748 seconds in the case of 1,000 requests in PSEV. Therefore, the messaging server response time typically takes between 0 and 3 seconds which proves the scalability of the solution.

## D. Memory and Power Consumption

As PSEV uses Blockchain and since the IoT devices have low computational capacities and require low power consumption, the consumption should be carefully assessed. In this study, the computational assessment was performed using a Huawei P8 Lite (2 GB Ram, a Hisilicon Kirin 620 Processor, Li-Po 2500 mAh battery). The operating system of this Huawei P8 Lite is Android 7.0 Nougat. Android runs on top of Linux kernel with a custom JVM on top of it and employs an innovative power management framework for saving power. Accordingly, a comparison with the available commercial applications should be conducted. Fig. 9 illustrates the testing results indicating that the energy consumption of the proposed solution of 25.43 mAh is comparable to Facebook and Skype applications that consume 18.56 mAh and 21.66 mAh respectively. In the other hand, the internal memory of the mobile devices is important for managing the processes, services, and applications that are in-built or installed. In general, applications with lower memory requirements are faster to run. The proposed solution consumes 102.4 MB which is lower than the most famous available applications. Fig. 9 depicts the memory consumption of Skype, WhatsApp, and Facebook which is 233 MB, 134 MB and 106 MB respectively.

## E. Integrity

For the systems that share sensitive data among the participants, it is critical to ensure integrity, as it is their fundamental feature. Accordingly, the assessment of the data integrity feature of the tested solution is crucial. Data integrity refers to how reliable and accurate is the data during the life cycle. Data security and integrity are highly interconnected. Uncorrupted data remains whole and unchanged in the complete state. Thus, to ensure security, data must be consistent during the entire life cycle. The following standards must be complied with for ensuring data reliability:

- The accuracy of data – data do not contain errors, and they are verified by the protocol.
- The data originality – sources can be accessed, and data preserve their initial form.
- Contemporary – data are being recorded at the moment of execution and observation.
- Legible – easy understanding, permanent recording, and preservation of original entries.

The role of cryptographic Hashing & Merkle Trees is to maintain the integrity of the data in the same form, regardless if the Blockchain is private or public. Merkle Trees [78] have the following main advantages. First, they have sufficient capacity to guarantee that the data are valid and with preserved integrity. Moreover, Merkel Trees require less disk space or memory because their proofs are prompt and easy to compute. Furthermore, to transmit the proofs across the networks and to manage them, minimal information is needed. In addition, cryptographic hashing maintains the data security and integrity recorded on Blockchain efficiently. Security is guaranteed through encryption, while signatures are updated every time when the data is modified, which ensures integrity. As the PSEV relies on Blockchain technology, data integrity is constantly ensured during the communication between vehicles.

## F. Consistency

Consistency is a critical criterion for the assessment of the new system. Consistency refers to assessments of the same project that generates comparable results when various raters use the identical method. The developed solution uses the Ethereum Blockchain to establish the consensus mechanism. Therefore, following [78], the reconciliation process is not required. The assumption behind the consistency mechanism is that the branch of most of the Proof-of-Work is real. A node must accept each block in the Blockchain, and the local copy of the database must be uncompromised to maintain consistency. The automatic resolution of the work occurs when nodes are in the temporary disagreement about the real consistent truths. It is impossible that honest nodes adapt to inconsistent chains in any case. Thus, the blocks located deeper in the chain within the network are constantly consistent. As Proof-of-Work prevents the unsolvable reconciliation process, the proposed PSEV system ensures consistency owing to the Blockchain.

## G. Confidentiality

Concerning computer systems, confidentiality refers to the notion that only authorized users can access sensitive and

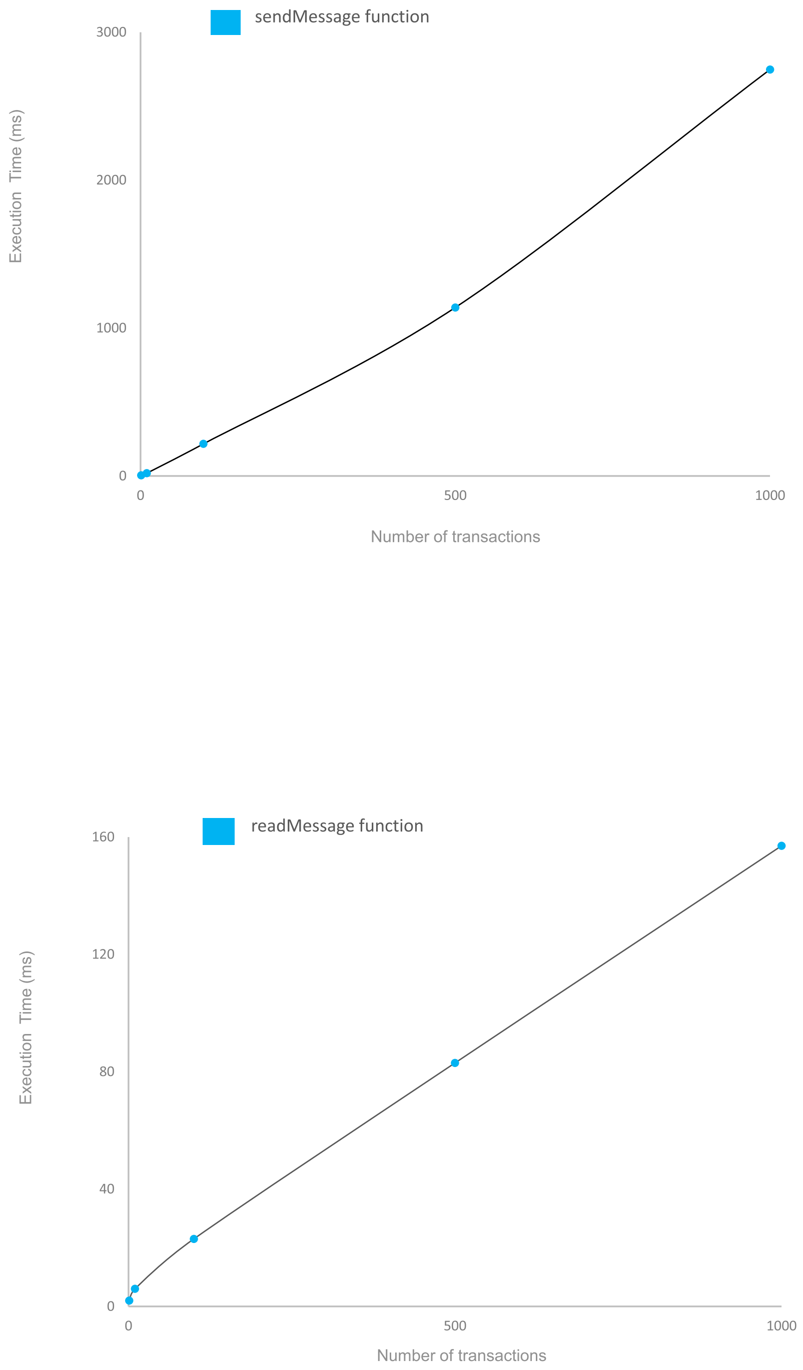


Fig. 8. Execution time of most called functions of PSEV-Communication.

protected data. Accordingly, specific mechanisms must be built to guarantee confidentiality and protect data from malicious attackers. In the context of Blockchain, confidentiality indicates that involved users can carry out a transaction, whereas other users cannot access specific information about the transaction. In the public Blockchain network, the notion of confidentiality is rejected, and accordingly, all the transactions are transparent. In contrast, private Blockchain ensures confidentiality to protect both the information on the transactions and the identities of participating nodes. The solution must satisfy the following requirements to achieve confidentiality:

- A person not participating in a given transaction cannot access the transaction details unless the participating parties share information.
- An unauthorized third party cannot access the identities of the participants of the transaction in a Blockchain until they share them.

### H. Immutability

Immutability prevents alteration after the creation. It would be needed to re-mine the blocks prior to the particular block to alter a transaction from history. In this particular case, it could

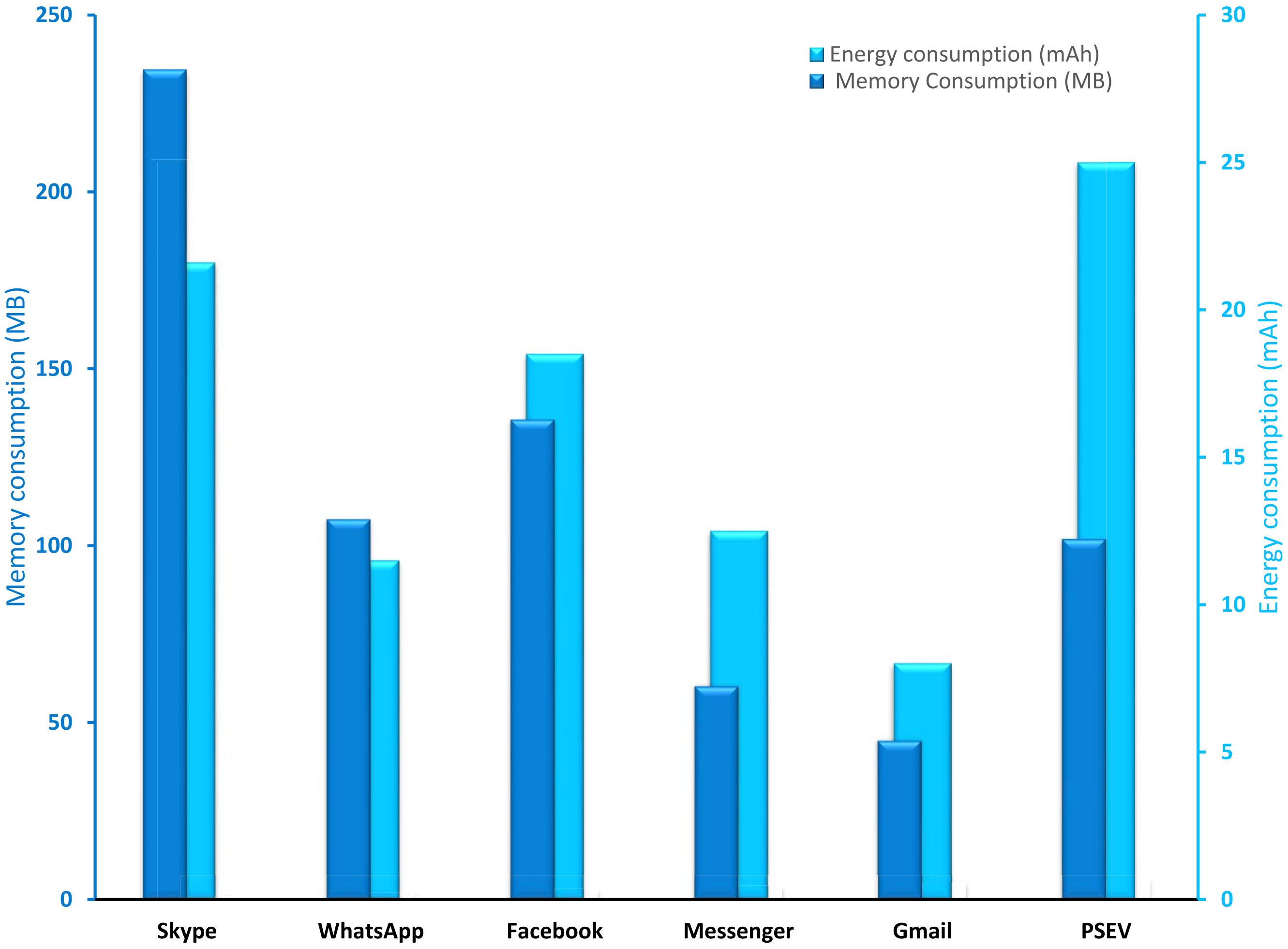


Fig. 9. The comparison of energy and memory consumption of PSEV and mobile applications commercially available.

be rejected in all ledger copies in the network. Also, it is necessary to reconstruct the Merkle tree of the block holding the transaction as well as the proof of work. Considering that the upcoming keeps the hash of this block that one would have to be re-mined as well. The upcoming block would need to be modified to include "previous block hash." Accordingly, a new block hash would be created. Re-mining of the block is required if the hash cannot meet the proper difficulty level. More precisely, the re-mining must be continued until the last block in the chain is re-mined. However, new blocks are added to the chain at the same moment when old blocks are re-mined and new simultaneously created blocks have to be re-mined as well. This action cannot be completed because of the needed computing power. Therefore, the proposed PSEV system ensures immutability.

## VI. Discussion

This paper presents the comparison between the developed solution PSEV and the similar solution DISV [49]. We selected DISV for comparison since it uses Ethereum and employs comparable tools (e.g., Ethereum GO [79], Truffle [77], and Ganache [80]). In this study, these solutions were tested under similar conditions in terms of costs, execution time, and memory and power consumption.

### A. Execution Time

One of the main criteria for evaluation of transportation management systems such as PSEV is execution time, as even minor delays can cause serious disruptions. In PSEV, it takes between 2 and 57 milliseconds to call any function, which proves that it is a real-time application. In DISV, the messaging server's response time takes between 2 and 245 milliseconds. Importantly, PSEV is able to send up to 1145 messages in 3 seconds, which outperforms similar solutions such as DISV that can send only 25 messages to all the participants during the same time. Furthermore, when comparing the two solutions, The call of functions of PSEV take less time then DISV.

### B. Cost

PSEV consists of three systems and each has particular functions. The Access Management System contains AddVehicles, GetVehiclesByAdress, AddParking, GetParkingByAddress, and GetAllParking functions. The Communication

Management System includes SendMessage and ReadMessage functions. Finally, the Parking Payment Management System encompasses SellTicket, GetNumberOfTicket, and BuyTicket functions. The most expensive is the generation and deployment of the prototype on the Blockchain ($30.7), followed by BuyTicket ($0.2739), SellTicket ($0.10613), AddParking ($0.09561), AddVehicles ( $0.08446), and SendMessage ($0.05677), while other functions do not incur costs. In contrast, DISV has only three functions: DeployContract ($0.07572 onetime), SetMessage ($0.0456), and GetMessage (0). Thus, the comparison revealed that deployment of the PSEV is more expensive because it contains more smart contracts and features. It is worth to mention that the increase of the cost of sending and receiving messages by 25% in PSEV has increased the capacity of sending messages from 25 to 1145 messages in 3 seconds.

### C. Memory and Power Consumption

The assessment of memory and power consumption is required considering the low computational capacities and power of IoT devices. PSEV consumes 25.43 mA of power, which is comparable to the power consumption of commercial applications such as Facebook, Skype, Gmail, Messenger, and WhatsApp. DISV consumes on average 23.4 mAh of power. It is important to mention that PSEV requires little higher power than DISV because it has more features. In conclusion, the comparison revealed that both solutions are real-time applications. Furthermore, PSEV has higher costs, memory and power consumption. However, this is because PSEV is a more complex solution with more functions to be executed as described in section III.

## VII. Conclusion

This paper proposed an innovative Blockchain framework for vehicle communication and parking payment (PSEV). It employs three primary layers that ensure communication and payments in the IoV using Blockchain. The proposed solution uses the smart contract prototype on the TESTNET Ropsten of Ethereum. The paper assessed the following critical features of the solution: cost, execution time, integrity, consistency, confidentiality, immutability, memory and power consumption. The main purpose of this assessment was to evaluate whether or not the Blockchain can be used as a platform for IoV communications and parking payment securely and effectively. The results indicate that the PSEV is a real-time application which is faster and more scalable than the existing used communication solutions. Moreover, PSEV successfully responded to the primary challenges of V2X communications, in particular to security, and scalability. Furthermore, the solution successfully facilitated the data sharing and cooperation among the participants in intelligent transportation systems, enhanced the Advanced Driver Assistance Systems (ADAS), and consequently, contributed to the overall security and safety of the transportation system.

## References

[1] F. Dahlqvist. (2018). *Growing Opportunities in the Internet of Things*. Accessed: Mar. 7, 2020. [Online]. Available: https://www.mckinsey.com/industries/private-equity-and-principal-investors/our-insights/growing-opportunities-in-the-internet-of-things

[2] F. Yang, S. Wang, J. Li, Z. Liu, and Q. Sun, "An overview of Internet of vehicles," *China Commun.*, vol. 11, no. 10, pp. 1–15, Oct. 2014.

[3] M. Saini, A. Alelaiwi, and A. E. Saddik, "How close are we to realizing a pragmatic VANET solution? A meta-survey," *ACM Comput. Surv.*, vol. 48, no. 2, pp. 1–40, Nov. 2015.

[4] S. F. Hasan, X. Ding, N. H. Siddique, and S. Chakraborty, "Measuring disruption in vehicular communications," *IEEE Trans. Veh. Technol.*, vol. 60, no. 1, pp. 148–159, Jan. 2011.

[5] B. Aslam, P. Wang, and C. C. Zou, "Extension of Internet access to VANET via satellite receive–only terminals," *Int. J. Ad Hoc Ubiquitous Comput.*, vol. 14, no. 3, pp. 172–190, 2013.

[6] J. Toutouh and E. Alba, "Light commodity devices for building vehicular ad hoc networks: An experimental study," *Ad Hoc Netw.*, vol. 37, pp. 499–511, Feb. 2016.

[7] S. Bitam, A. Mellouk, and S. Zeadally, "VANET-cloud: A generic cloud computing model for vehicular ad hoc networks," *IEEE Wireless Commun.*, vol. 22, no. 1, pp. 96–102, Feb. 2015.

[8] D. Mohr *et al.*, "The road to 2020 and beyond: What's driving the global automotive industry," *McKinsey Automot Assem. Latest Think*, vol. 28, no. 3, p. 2014, 2013.

[9] O. Kaiwartya *et al.*, "Internet of vehicles: Motivation, layered architecture, network model, challenges, and future aspects," *IEEE Access*, vol. 4, pp. 5356–5373, 2016.

[10] M. S. Sheikh, J. Liang, and W. Wang, "A survey of security services, attacks, and applications for vehicular ad hoc networks (VANETs)," *Sensors*, vol. 19, no. 16, p. 3589, Aug. 2019.

[11] F. Ahmad, J. Hall, A. Adnane, and V. N. L. Franqueira, "Faith in vehicles: A set of evaluation criteria for trust management in vehicular ad-hoc network," in *Proc. IEEE Int. Conf. Internet Things*, Jun. 2017, pp. 44–52.

[12] M. Aloqaily, S. Otoum, I. A. Ridhawi, and Y. Jararweh, "An intrusion detection system for connected vehicles in smart cities," *Ad Hoc Netw.*, vol. 90, Jul. 2019, Art. no. 101842.

[13] T. Hunt. (2016). *Controlling Vehicle Features of Nissan LEAFs Across the Globe Via Vulnerable*. Accessed: Mar. 7, 2020. [Online]. Available: https://www.troyhunt.com/controlling-vehicle-features-of-nissan/

[14] P. Paganini. (2016). *Hackers Can Remotely Disable Car Alarm on Mitsubishi Outlander PHEV SUVs*. Accessed: Mar. 7, 2020. [Online]. Available: https://securityaffairs.co/wordpress/48114/hacking/mitsubishi-outlander-phev-hacking.html/

[15] J. Finkle. (2016). *Tesla Fixes Security Bugs After Claims of Model S Hack*. Accessed: Mar. 7, 2020. [Online]. Available: https://www.reuters.com/article/us-tesla-cyber/tesla-fixes-security-bugs-after-claims-of-model-s-hack-idUSKCN11Q2SD/

[16] T. M. Fernández-Caramés and P. Fraga-Lamas, "A review on the use of blockchain for the Internet of Things," *IEEE Access*, vol. 6, pp. 32979–33001, 2018.

[17] A. Reyna, C. Martín, J. Chen, E. Soler, and M. Díaz, "On blockchain and its integration with IoT. Challenges and opportunities," *Future Gener. Comput. Syst.*, vol. 88, pp. 173–190, Nov. 2018.

[18] M. Banerjee, J. Lee, and K.-K.-R. Choo, "A blockchain future for Internet of Things security: A position paper," *Digit. Commun. Netw.*, vol. 4, no. 3, pp. 149–160, Aug. 2018.

[19] G. Zyskind, O. Nathan, and A. S'. Pentland, "Decentralizing privacy: Using blockchain to protect personal data," in *Proc. IEEE Secur. Privacy Workshops*, May 2015, pp. 180–184.

[20] N. Kshetri, "Blockchain's roles in strengthening cybersecurity and protecting privacy," *Telecommun. Policy*, vol. 41, no. 10, pp. 1027–1038, Nov. 2017.

[21] H. Zhao, Y. Zhang, Y. Peng, and R. Xu, "Lightweight backup and efficient recovery scheme for health blockchain keys," in *Proc. IEEE 13th Int. Symp. Auto. Decentralized Syst. (ISADS)*, Mar. 2017, pp. 229–234.

[22] X. Yue, H. Wang, D. Jin, M. Li, and W. Jiang, "Healthcare data gateways: Found healthcare intelligence on blockchain with novel privacy risk control," *J. Med. Syst.*, vol. 40, no. 10, pp. 1–8, Oct. 2016.

[23] P. Sylim, F. Liu, A. Marcelo, and P. Fontelo, "Blockchain technology for detecting falsified and substandard drugs in distribution: Pharmaceutical supply chain intervention," *JMIR Res. Protocols*, vol. 7, no. 9, Sep. 2018, Art. no. e10163.

[24] J. Dai and M. A. Vasarhelyi, "Toward blockchain-based accounting and assurance," *J. Inf. Syst.*, vol. 31, no. 3, pp. 5–21, Sep. 2017.

[25] B. Wang, J. Sun, Y. He, D. Pang, and N. Lu, "Large-scale election based on blockchain," *Procedia Comput. Sci.*, vol. 129, pp. 234–237, Sep. 2018.

[26] R. Osgood, "The future of democracy: Blockchain voting," *Inf. Secur.*, vol. 14, pp. 1–21, Dec. 2016.

[27] S. Ølnes, J. Ubacht, and M. Janssen, "Blockchain in government: Benefits and implications of distributed ledger technology for information sharing," *Government Inf. Quart.*, vol. 34, no. 3, pp. 355–364, 2017.

[28] X. Liang, J. Zhao, S. Shetty, J. Liu, and D. Li, "Integrating blockchain for data sharing and collaboration in mobile healthcare applications," in *Proc. IEEE 28th Annu. Int. Symp. Pers., Indoor, Mobile Radio Commun. (PIMRC)*, Oct. 2017, pp. 1–5.

[29] V. L. Lemieux, "In blockchain we trust? Blockchain technology for identity management and privacy protection," in *Proc. Conf. E-Democracy Open Government*, 2017, p. 57.

[30] A. Dorri, M. Steger, S. S. Kanhere, and R. Jurdak, "BlockChain: A distributed solution to automotive security and privacy," *IEEE Commun. Mag.*, vol. 55, no. 12, pp. 119–125, Dec. 2017.

[31] B. Bhushan, A. Khamparia, K. M. Sagayam, S. K. Sharma, M. A. Ahad, and N. C. Debnath, "Blockchain for smart cities: A review of architectures, integration trends and future research directions," *Sustain. Cities Soc.*, vol. 61, Oct. 2020, Art. no. 102360.

[32] B. Bhushan, C. Sahoo, P. Sinha, and A. Khamparia, "Unification of Blockchain and Internet of Things (BIoT): Requirements, working model, challenges and future directions," *Wireless Netw.*, vol. 27, pp. 55–90, Jan. 2020.

[33] Z. Zheng, S. Xie, H.-N. Dai, X. Chen, and H. Wang, "Blockchain challenges and opportunities: A survey," *Int. J. Web Grid Services*, vol. 14, no. 4, pp. 352–375, 2018.

[34] S. Nakamoto and A. Bitcoin. (2020). *A Peer-to-Peer Electronic Cash-system*. Bitcoin. [Online]. Available: https://bitcoin.org/bitcoin.pdf

[35] X. Huang, C. Xu, P. Wang, and H. Liu, "LNSC: A security model for electric vehicle and charging pile management based on blockchain ecosystem," *IEEE Access*, vol. 6, pp. 13565–13574, 2018.

[36] J. Kang, R. Yu, X. Huang, S. Maharjan, Y. Zhang, and E. Hossain, "Enabling localized peer-to-peer electricity trading among plug-in hybrid electric vehicles using consortium blockchains," *IEEE Trans. Ind. Informat.*, vol. 13, no. 6, pp. 3154–3164, Dec. 2017.

[37] L. Li, J. Liu, L. Cheng, S. Qiu, W. Wang, and X. Zhang, "CreditCoin: A privacy-preserving blockchain-based incentive announcement network for communications of smart vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 7, pp. 2204–2220, Jul. 2018.

[38] Z. Yang, K. Zheng, K. Yang, and V. C. M. Leung, "A blockchain-based reputation system for data credibility assessment in vehicular networks," in *Proc. IEEE 28th Annu. Int. Symp. Pers., Indoor, Mobile Radio Commun. (PIMRC)*, Oct. 2017, pp. 1–5.

[39] Y. Yuan and F.-Y. Wang, "Towards blockchain-based intelligent transportation systems," in *Proc. IEEE 19th Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2016, pp. 2663–2668.

[40] J. Kang *et al.*, "Blockchain for secure and efficient data sharing in vehicular edge computing and networks," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4660–4670, Jun. 2019.

[41] B. Leiding, P. Memarmoshrefi, and D. Hogrefe, "Self-managed and blockchain-based vehicular ad-hoc networks," in *Proc. ACM Int. Joint Conf. Pervas. Ubiquitous Comput., Adjunct*, Sep. 2016, pp. 137–140.

[42] E. Reilly, M. Maloney, M. Siegel, and G. Falco, "An IoT integrity-first communication protocol via an ethereum blockchain light client," in *Proc. IEEE/ACM 1st Int. Workshop Softw. Eng. Res. Practices Internet Things*, May 2019, pp. 15–19.

[43] G. Falco, C. Li, P. Fedorov, C. Caldera, R. Arora, and K. Jackson, "NeuroMesh: IoT security enabled by a blockchain powered botnet vaccine," in *Proc. Int. Conf. Omni-Layer Intell. Syst.*, May 2019, pp. 1–6.

[44] G. Falco and J. E. Siegel, "A distributed 'Black Box' audit trail design specification for connected and automated vehicle data and software assurance," 2020, *arXiv:2002.02780*. [Online]. Available: http://arxiv.org/abs/2002.02780

[45] U. Javaid, M. N. Aman, and B. Sikdar, "A scalable protocol for driving trust management in Internet of vehicles with blockchain," *IEEE Internet Things J.*, vol. 7, no. 12, pp. 11815–11829, Dec. 2020.

[46] U. Javaid, M. N. Aman, and B. Sikdar, "DrivMan: Driving trust management and data sharing in VANETs with blockchain and smart contracts," in *Proc. IEEE 89th Veh. Technol. Conf. (VTC-Spring)*, Apr. 2019, pp. 1–5.

[47] M. N. Aman, U. Javaid, and B. Sikdar, "A privacy-preserving and scalable authentication protocol for the Internet of vehicles," *IEEE Internet Things J.*, vol. 8, no. 2, pp. 1123–1139, Jan. 2021.

[48] S. Rowan, M. Clear, M. Gerla, M. Huggard, and C. Mc Goldrick, "Securing vehicle to vehicle communications using blockchain through visible light and acoustic side-channels," 2017, *arXiv:1704.02553*. [Online]. Available: http://arxiv.org/abs/1704.02553

[49] R. Jabbar, M. Kharbeche, K. Al-Khalifa, M. Krichen, and K. Barkaoui, "Blockchain for the Internet of vehicles: A decentralized IoT solution for vehicles communication using ethereum," *Sensors*, vol. 20, no. 14, p. 3928, Jul. 2020.

[50] R. Jabbar, M. Krichen, M. Kharbeche, N. Fetais, and K. Barkaoui, "A model-based testing framework for validating an IoT solution for blockchain-based vehicles communication," hal.archives-ouvertes, Lyon, France, Tech. Rep. hal-02513237, 2020.

[51] R. Jabbar, M. Krichen, M. Kharbeche, N. Fetais, and K. Barkaoui, "Un cadre de test formel pour la validation d'un système de communication inter-véhiculaire basésur les IOTs et la blockchain," hal.archives-ouvertes, Lyon, France, Tech. Rep. hal-02513235, version 1, 2020.

[52] R. Jabbar, M. Krichen, M. Kharbeche, N. Fetais, and K. Barkaoui, "A formal model-based testing framework for validating an IoT solution for blockchain-based vehicles communication," in *Proc. 15th Int. Conf. Eval. Novel Approaches Softw. Eng.*, 2020, pp. 595–602.

[53] R. Jabbar, K. Al-Khalifa, M. Kharbeche, W. Alhajyaseen, M. Jafari, and S. Jiang, "Applied Internet of Things IoT: Car monitoring system for modeling of road safety and traffic system in the state of qatar," in *Proc. Qatar Found. Annu. Res. Conf. Process.*, 2018, Art. no. ICTPP1072.

[54] R. Jabbar, M. Shinoy, M. Kharbeche, K. Al-Khalifa, M. Krichen, and K. Barkaoui, "Urban traffic monitoring and modeling system: An IoT solution for enhancing road safety," in *Proc. Int. Conf. Internet Things, Embedded Syst. Commun. (IINTEC)*, Dec. 2019, pp. 13–18.

[55] R. Jabbar, K. Al-Khalifa, M. Kharbeche, W. Alhajyaseen, M. Jafari, and S. Jiang, "Real-time driver drowsiness detection for Android application using deep neural networks techniques," *Procedia Comput. Sci.*, vol. 130, pp. 400–407, Sep. 2018.

[56] R. Jabbar, M. Shinoy, M. Kharbeche, K. Al-Khalifa, M. Krichen, and K. Barkaoui, "Driver drowsiness detection model using convolutional neural networks techniques for Android application," 2020, *arXiv:2002.03728*. [Online]. Available: http://arxiv.org/abs/2002.03728

[57] (2021). *Ethereum*. Accessed: Feb. 20, 2021. [Online]. Available: https://ethereum.org/

[58] (2020). *All Cryptocurrencies*. Accessed: Mar. 7, 2020. [Online]. Available: https://coinmarketcap.com/currencies/bitcoin/

[59] J. Poon and T. Dryja, "The bitcoin lightning network: Scalable off-chain instant payments," Satoshi Nakamoto Inst., Tech. Rep., 2016.

[60] R. Network-Fast, "Cheap, scalable token transfers for Ethereum," Raiden, Mainz, Germany, Tech. Rep. 7, 2018.

[61] S. Thomas and E. Schwartz. (2015). *A Protocol for Interledger Payments*. [Online]. Available: https://interledger.org/interledger.pdf

[62] M. Herlihy, "Atomic cross-chain swaps," in *Proc. ACM Symp. Princ. Distrib. Comput.*, Jul. 2018, pp. 245–254.

[63] G. Malavolta, P. Moreno-Sanchez, A. Kate, M. Maffei, and S. Ravi, "Concurrency and privacy with payment-channel networks," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2017, pp. 455–471.

[64] P. Prihodko, S. Zhigulin, M. Sahno, A. Ostrovskiy, and O. Osuntokun, "Flare: An approach to routing in lightning network," Indian Inst. Technol. Bomby, Mumbai, India, Dual Degree Stage Rep. 2018, 2016.

[65] R. Lu, X. Lin, H. Zhu, and X. Shen, "SPARK: A new VANET-based smart parking scheme for large parking lots," in *Proc. 28th Conf. Comput. Commun.*, Apr. 2009, pp. 1413–1421.

[66] L. Zhu, M. Li, Z. Zhang, and Z. Qin, "ASAP: An anonymous smart-parking and payment scheme in vehicular networks," *IEEE Trans. Dependable Secure Comput.*, vol. 17, no. 4, pp. 703–715, Jul. 2020.

[67] S. Mathur *et al.*, "ParkNet: Drive-by sensing of road-side parking statistics," in *Proc. 8th Int. Conf. Mobile Syst., Appl., Services*, 2010, pp. 123–136.

[68] J. Ni, K. Zhang, X. Lin, Y. Yu, and X. Shen, "Cloud-based privacy-preserving parking navigation through vehicular communications," in *Security and Privacy in Communication Networks. SecureComm* (Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering), vol. 198, R. Deng, J. Weng, K. Ren, and V. Yegneswaran, Eds. Cham, Switzerland: Springer, 2017, doi: 10.1007/978-3-319-59608-2_5.

[69] J. Mola. (2019). *PARQ—Green, Smart and Connected City Platform*. Accessed: Mar. 7, 2020. [Online]. Available: https://ion.community/topic/4047/parq-a-green-smart-and-connected-city-platform

[70] P. Team. (2018). *PARQ—Green, Smart and Connected City Platform*. Accessed: Mar. 7, 2020. [Online]. Available: https://parkres.org/Parkreswhitepaper.pdf

[71] (2020). *Drowsy Driving NHTSA Reports*. [Online]. Available: https://www.nhtsa.gov/risky-driving/drowsy-driving

[72] Z. Guan *et al.*, "APPA: An anonymous and privacy preserving data aggregation scheme for fog-enhanced IoT," *J. Netw. Comput. Appl.*, vol. 125, pp. 82–92, Jan. 2019.

[73] A. Lahbib, K. Toumi, A. Laouiti, A. Laube, and S. Martin, "Blockchain based trust management mechanism for IoT," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Apr. 2019, pp. 1–8.

[74] (2020). *MythX: Smart Contract Security Service for Ethereum*. [Online]. Available: https://mythx.io/

[75] (2020). *Open Web Application Security Project (OWASP)*. [Online]. Available: https://www.owasp.org/index.php/Main_Page

[76] (2020). *Smart Contract Security Verification Standard*. [Online]. Available: https://github.com/securing/SCSVS

[77] (2020). *Truffle Suite: Sweet Tools for Smart Contracts*. [Online]. Available: https://www.trufflesuite.com/

[78] M. Löf, "Decentralized transactions in a centralized environment: A blockchain study within the transport industry," M.S. thesis, School Comput. Sci. Commun. (CSC), Stockholm, Sweden, 2017.

[79] (2020). *Ethereum/go-Ethereum*. [Online]. Available: https://github.com/ethereum/go-ethereum

[80] (2020). *Ganache|Truffle Suite*. [Online]. Available: https://www.trufflesuite.com/

**Rateb Jabbar** received the M.S. degree in software engineering from the University of Tunisia in 2012. He worked as a senior software engineer for eight years specialist in Web, Cloud, machine learning, and Blockchain technologies. He is a Research Assistant with the Department of Computer Science and Engineering, Qatar University. He is a Microsoft Certified Professional and a Microsoft Certified Technology Specialist in Developing ASP.NET MVC 4 Web Applications, Microsoft Azure cloud service, and Web Services.

**Noora Fetais** (Senior Member, IEEE) received the Ph.D. degree from the University of Sussex, U.K. She is an Assistant Professor with the Department of Computer Science and Engineering, Qatar University. Her expertise is in visualization for cybersecurity. She was a member of the first batch of Qatar Leadership Center (Current and Future Leaders Program 2011–2013). She is currently joining the first batch of QLC Executive Master in Leadership, Policy and Innovation at Georgetown University. She is also holding various professional positions including; the Vice Chair of IEEE-Qatar Section, Qatar Ambassador of Women in Data Science (WiDS), Stanford University, among others. She was the first Women to Chair the Faculty Senate of Qatar University in 2016 and the Director of KINDI Center for Computing Research, Qatar University (QU) from 2017 to 2020. She is a member of Qatar-U.K. Alumni Networking, and the Board of Advisors for Qatar Academy is part of the Qatar Foundation (QF) for Education, Science and Community Development.

**Mohamed Kharbeche** (Member, IEEE) received the M.Sc. and Ph.D. degrees in operations research from the Higher Institute of Management, Tunisia. He is currently a Research Associate with the Qatar Transportation and Traffic Safety Center, Qatar University, and working on several areas on traffic safety, transportation, and road user behavior. He is also a principal investigator of many research projects on pedestrian crossing behavior, traffic system modeling and simulation, drowsiness detection, factors affecting driver reaction time using naturalistic driving data, and hazardous material transportation funded by Qatar National Research Fund (QNRF). He has authored or coauthored more than 50 journals and conferences. He is also leading action plans of the National Traffic Safety Strategy 2013–2022 and a project on crash experiences and safety perspectives for motorcyclists in Qatar. He managed several projects sponsored by ExxonMobil Qatar, to evaluate school bus transportation and safety around schools as well as pedestrian accidents in Qatar.

**Moez Krichen** (Member, IEEE) received the Ph.D. degree in computer science from the University of Joseph Fourrier, Grenoble, France, in 2007, and the HDR (Ability to Conduct Researches) degree in computer science from the University of Sfax, Sfax, Tunisia, in 2018. He is currently an Assistant Professor with the National School of Engineers of Sfax and a member of the Research Laboratory on Development and Control of Distributed Applications - REDCAD, Sfax. His research interests include model-based conformance, load and security testing methodologies for real-time, distributed, and dynamically adaptable systems. Moreover, he works on applying formal methods to several modern technologies, such as smart cities, the Internet of Things (IoT), smart vehicles, drones, and healthcare systems. He is also working on formal aspects related to deep learning, data mining, blockchain, smart contracts, and optimization.

**Kamel Barkaoui** is a Full Professor with the Conservatoire National des Arts et Métiers (Le Cnam), Paris. He leaded or participated in more than ten international research projects. His research interests include formal methods for verification, control, and performance evaluation of concurrent and distributed systems. He received the Outstanding Paper Award at the IEEE International Conference on System Man and Cybernetics (Vancouver 1995). He has been a recipient of the Prime d'Excellence Scientifique since 1998. He is the SC Chair of the International Conference on Verification and Evaluation of Computer and Communication Systems (VECoS). He was the General Co-Chair of the 18th International Symposium on Formal Methods (FM 2012) and the General Chair of 35th International Conference on Application and Theory of Petri Nets and Concurrency (Petri Nets 2014) and the 14th International Conference on Application of Concurrency to System Design (ACSD 2014).

**Mohammed Shinoy** received the master's degree in mechanical and mechatronics from the University of Waterloo, Canada. He is a passionate about Machine Learning and holds a nanodegree in the same. He is a Research Assistant with the Qatar Transportation and Traffic Safety Center, College of Engineering, Qatar University.