Consortium Blockchain for Cooperative Location Privacy Preservation in 5G-enabled Vehicular Fog Computing

Abdelwahab Boualouache, *Member*, *IEEE*, Hichem Sedjelmaci, *Member*, *IEEE*, and Thomas Engel

***Abstract*—Privacy is a key requirement for connected vehicles. Cooperation between vehicles is mandatory for achieving location privacy preservation. However, non-cooperative vehicles can be a big issue to achieve this objective. To this end, we propose a novel monetary incentive scheme for cooperative location privacy preservation in 5G-enabled Vehicular Fog Computing. This scheme leverages a consortium blockchain-enabled fog layer and smart contracts to ensure a trusted and secure cooperative Pseudonym Changing Processes (PCPs). We also propose optimized smart contracts to reduce the monetary costs of vehicles while providing more location privacy preservation. Moreover, a resilient and lightweight Utility-based Delegated Byzantine Fault Tolerance (U-DBFT) consensus protocol is proposed to ensure fast and reliable block mining and validation. The performance analysis shows that our scheme has effective incentive techniques to stimulate non-cooperative vehicles and provides optimal monetary cost management and secure, private, fast validation of blocks.**

***Index Terms*—5G-enabled Vehicular Fog Computing; Consortium Blockchain; Location Privacy; Incentive Mechanisms; Pseudonym Changing;**

I. Introduction

5G communication technologies are expected to revolutionize Intelligent Transportation Systems (ITSs). Vehicular networks are the main component of ITS that is taking part in this revolution to enable high bandwidth and ultra-low latency for 5G-enabled V2X services [1]. Empowered by fog computing paradigm, 5G-enabled Vehicular Fog Computing (5GVFC) is addressing limitations of traditional vehicular networks in terms of latency, improving safety, mobility, and driver experience during journeys [2]. 5GVFC has already demonstrated its benefits in several V2X domains such as task offloading [3], data caching [4], data collection [5], and data sharing [6] and it is also envisioned to support V2X security and privacy services. Inherited from traditional vehicular networks, location privacy is still a complex issue in 5GVFC. 5G-enabled V2X services and applications such as collision avoidance, cooperative driving, and traffic management rely on the periodic broadcast of safety-related messages, known as beacons. These beacons are aiming at establishing cooperative awareness between vehicles [7]. However, these beacons carry sensitive information such as position, speed, velocity, and heading, which may threaten the location privacy of vehicles' users. Indeed, these messages could easily be eavesdropped by a passive adversary who can link these messages with their corresponding vehicles' identifiers and track the trajectory of the vehicle during its journey, which violates the location privacy of drivers [8]. One solution to avoid tracking vehicles from their transmitted beacons is the use of multiple temporary identifiers, called pseudonyms. Vehicles periodically change their pseudonyms to achieve the unlikeability between their beacons. This solution, called the Pseudonym-Changing approach, is already part of vehicular security standards [9, 10]. However, if only one vehicle changes its pseudonym, the attacker can easily link between its pseudonyms due to pseudonym syntactic linking attacks [11]. A set of strategies has been proposed to ensure cooperation between vehicles in Pseudonym Changing Processes (PCPs) [12]. However, these strategies fail to provide the required protection with the presence of non-cooperative vehicles, which tend not to get involved in PCPs. This non-cooperative behavior is mainly due to the rationality and the selfishness of vehicles that aim at increasing their privacy levels and saving their pseudonyms to avoid the costs generated from requesting new pseudonym sets.

Over the past few years, some incentive mechanisms have been proposed to stimulate vehicles to cooperate in PCPs. These mechanisms can be classified according to the used incentive tool into two categories: (i) game theory-based incentive mechanisms [13–15], and (ii) reputation-based mechanisms [16, 17]. However, in game theory-based incentive mechanisms, vehicles cooperate only if the payoffs are greater than costs. Also, the complexity of the game-theoretical system increase with the number of players in the system [18]. Indeed, the Nash equilibrium solution is usually hard to achieve due to the dynamic properties that characterize vehicular networks. On the other hand, reputation-based mechanisms including centralized and distributed mechanisms, are an easy target for internal attackers, which can exploit the stored reputation values for their self-interests. These mechanisms are also vulnerable to denial of service attacks aiming at breaking the reputation system. Besides this, the two categories propose non-monetary incentive mechanisms, which make it difficult for vehicles to benefit from cooperation for recovering pseudonym-changing costs. Blockchain technology has recently emerged to enable secure transactions among distributed entities through the use of an immutable ledger, cryptocurrency, and the execution of smart contracts [19]. Entities are executing a consensus

A. Boualouache and T. Engel are with University of Luxembourg, Esch-sur-Alzette, AVE, 4365, Luxembourg e-mail: ({abdelwahab.boualouache and thomas.engel}@uni.lu)

H. Sedjelmaci is with Orange Labs, 44 Avenue de la République, 92320 Châtillon, France e-mail: hichem.sedjelmaci@orange.com

protocol for validation of transactions, generation of blocks, and building of hash chain over blocks [20]. Few schemes have recently used this technology for pseudonyms shuttling and revocation processes in vehicular networks [21, 22]. However, none of these schemes have exploited blockchain to support PCPs. Also, these schemes use the proof-of-work (POW) consensus protocol, which is proved that it wastes a lot of energy and has slow validation of transactions in blockchain systems [23], [24].

In this paper, to address the aforementioned issues, we propose a novel monetary incentive scheme for cooperative location privacy in 5G-enabled vehicular fog computing. This scheme relies on consortium blockchain deployed in a fog layer and on smart contracts to achieve a trusted and secure cooperation between vehicles in PCPs. Optimized smart contracts are also proposed to reduce the monetary costs of vehicles while providing more location privacy preservation. Our scheme leverages a resilient Utility-based Delegated Byzantine Fault Tolerance (U-DBFT) consensus protocol to ensure fast and reliable block mining and validation. Simulation and analytic results demonstrate that the proposed scheme is effective to ensure successful and secure PCPs.

The main contributions of this paper can then be summarized as follows:

- We propose a secure and privacy-preserving architecture for cooperative location privacy preservation between vehicles in 5G-enabled vehicular fog computing.
- We design one-to-many smart contracts to achieve trusted and secure cooperation between a pseudonym changing requestor and a set of pseudonym changing cooperators in PCPs.
- Leveraging k-means clustering algorithm, we propose optimized smart contracts aiming at establishing many-to-many smart contracts between a set of pseudonym changing requestors and a set of pseudonym changing cooperators for reducing the monetary costs of vehicles while providing more location privacy preservation.
- We propose an efficient U-DBFT consensus protocol that provides fast consensus rounds and high resilience to faulty and malicious nodes.
- We carry out a set of simulations and analytic evaluations for evaluating the cooperative behavior compared to existing cooperation strategies and for performing monetary and blockchain analysis using both standard smart contacts and optimized smart contracts. We also formulate a non-cooperative security game to capture different malicious behaviors in the proposed scheme.

The remainder of this paper is organized as follows. Related works are described in Section II. The proposed architecture for cooperative location privacy preservation is presented in Section III. Section IV describes the designed smart contract for cooperative pseudonym changing. Section V presents optimized smart contracts. The U-DBFT consensus protocol is described in Section VI. The results of the performance evaluation are presented in Section VII. Section VIII discusses the obtained results and performs a security analysis of our scheme. A conclusion is given in Section IX.

II. Related Work

A. Pseudonym Changing strategies

During the last few years, several Pseudonym Changing Strategies (PCSs) have been proposed to prevent pseudonym linking attacks. These strategies are classified into two categories [12]: (i) Mix-zone-based strategies, and (ii) Mix-context-based strategies. In the former category, the PCP only occurs on predefined road areas, called mix-zones. We mention as examples of these strategies: (i) *Freudiger et al. (a)* [25], which proposed to install Cryptographic Mix (CMIX) zones on road intersections where all safety messages are encrypted, (ii) *Lu et al.* [15], which proposed to perform PCPs at Social Spots such as signalized intersections and parking lots, and (iii) *Boualouache and al. (a)* [26], which proposed to stop broadcasting safety messages at signalized intersections only while the traffic light is red. On the other hand, mix-context-based strategies can occur everywhere, and whenever the predefined context is found. We mention as examples of these strategies: *Gerlach et al.* [27], which proposed that a vehicle changes its pseudonym only if it detects k neighboring vehicles at a distance smaller than the minimal distance and has a similar direction with it within its communication range, (ii) *Wasef et al.* [28], which introduced Random Encryption Periods (REPs). When a vehicle decides to change its pseudonym, it sends a request to its neighbors for starting a REP. During a REP, safety messages are encrypted using a shared group key, and (iii) *Boualouache et al. (b)* [11], which proposed the Traffic-aware PCS, where vehicles continuously monitor the road traffic status to find optimal locations where the silent mix zone (SM) can be established. However, although the important number of strategies that have been proposed, only a few of them propose incentive mechanisms to stimulate non-cooperative vehicles to participate in PCPs. These incentive mechanisms are described in the next section.

B. Incentive mechanisms of PCS

Incentive mechanisms for PCPs can be classified into the two following categories:

Game theory-based incentive mechanisms: *Lu et al.* [15] demonstrated the feasibility of Social Spots strategy using a simplified game-theoretic to demonstrate the feasibility of the proposed strategy assuming that all vehicles are rational. *Freudiger et al.* [13] proposed a game-theoretical model that takes into account each vehicle's gained payoff and the costs to decide whether to cooperate or no in PCP. *Du et al.* [14] proposed the AVATAR scheme that generates a number of virtual nodes in the proximity of a node and allows both virtual and real nodes to make a coordinated PCP. A reward mechanism based on a multiunit discriminatory auction game is also proposed to stimulate each node to participate in PCPs. However, in these mechanisms, vehicles cooperate only if the payoffs are greater than costs. Also, the complexity of the game-theoretical system increase with the number of players in the system [18]. Indeed, the Nash equilibrium solution is usually hard to achieve due to the dynamic properties that characterize vehicular networks.

Reputation-based incentive mechanisms: *Ying et al.* [16] proposed that each vehicle establishes its mix zone with the assistance of third trusted units, called Control Servers (CSs). A reputation-based mechanism is also proposed to stimulate non-cooperative vehicles to cooperate in other vehicles' mix zones. The reputation value of each vehicle is maintained by CSs, which increases it each time the vehicle cooperates in a mix-zone of another vehicle. The accumulated reputation value is used as a credit when a vehicle requests to create its mix-zone. The authors in [17] also proposed a reputation-based mechanism to motivate rational vehicles to enter Vehicular Location Privacy Zones (VLPZs). These zones are dedicated to perform PCPs and are managed by public or private organizations. Vehicles with low privacy levels can access VLPZs only their reputation values are above or equal to a certain threshold. VLPZs frequently sends invitations to vehicles to increase the number of vehicles inside them. The reputation value of a vehicle will be increased or decreased depending on whether the vehicle accepts or refuses the invitation respectively. However, reputation-based mechanisms including centralized and distributed mechanisms, are an easy target for internal attackers, which can exploit the stored reputation values for their self-interests. These mechanisms are also vulnerable to denial of service attacks aiming at breaking the reputation system.

C. Blockchain

Blockchain is a distributed and immutably distributed ledger to enable secure transactions among distributed entities [20]. There are two types of blockchain structures: public blockchain and consortium blockchain. While in the public blockchain every entity can build and verify blocks, in the consortium blockchain only a group of authorized members can to this. Unlike the public blockchain, the consortium blockchain is more suitable for energy-constrained and delay-sensitive networks with low energy and time consumption for achieving consensus [29]. Moreover, smart contracts (SCs) are scripts or programs residing on the blockchain in which the execution results are verified by miners. Their deployments and executions are triggered by users through transactions [30]. Recently several studies have proposed blockchain-based solutions to secure vehicular edge computing [31]. Li et al [32] proposed, CreditCoin, a privacy-preserving incentive announcement scheme based on blockchain to secure vehicular communications. Zhang and Chen. [33] proposed a data security sharing and storage system based on the consortium blockchain. Wang and al. [34] propose a secure charging system for electric vehicles based on smart SCs and consortium blockchain. Wang et al. [35] proposed, Parkingchain, a permissioned vehicular blockchain for secure and efficient resource sharing in vehicular edge computing (VEC) consisting of parked vehicles (PVs). The authors of [21, 22] exploit blockchain to propose a shuttle and revocation scheme of pseudonym sets. The blockchain network is composed of entities called Pseudonym Manager (PMs). Each PM receives expired pseudonym sets from vehicles and packages and broadcasts them in the blockchain network. Each PM uses its algorithm to shuffle the pseudonym sets and the obtain the results are added block. The proof of work (PoW) consensus protocol is used to determine the mining node that inserts the block into the ledger. Once the consensus is achieved, the public key infrastructure updates the links between the real identifiers of vehicles and their correspondent pseudonym sets and publishes them into the blockchain. However, none of these schemes have exploited blockchain to support cooperative PCPs. In addition, the POW consensus protocol wastes a lot of energy and has slow validation of transactions in blockchain systems.

In Table I, we compare our scheme with relevant state-of-the-art schemes. As we can in Table I (a), unlike location privacy incentive schemes [13, 14, 16, 17], our scheme exploits blockchain technology and smart contracts to provide secure cooperation between vehicles in PCPs. In addition, our scheme adopts a monetized realistic approach, which allows vehicles exploiting payments gained from cooperation for covering pseudonym changing costs. Moreover, our proposed scheme is generic .i.e it is independent of the applied pseudonym changing strategy. On the other hand, unlike [21, 22] presented in Table I (b), our scheme exploits the blockchain technology to support pseudonym changing strategies. In addition, our scheme uses a lightweight consensus protocol that consumes less energy than PoW.

TABLE I: A comparison of our scheme with relevant state-of-the-art schemes.

(a)

Solution	Incentive Mechanism	Secure	Monetized	Generic
Freudiger et al. [13]	Game Theory			X
AVATAR [14]	Game Theory			
MPSVLP [16]	Reputation			
PRIVANET [17]	Reputation			
Our Scheme	**Smart Contract**	**X**	**X**	**X**

(b)

Solution	Goal	Consensus Protocol
Bao et al.[21]	Pseudonym management	PoW
Lei et al. [22]	Pseudonym revocation	PoW
Our Scheme	Pseudonym charging and incentive scheme	U-DBFT

III. Blockchain-based architecture for cooperative location privacy preservation

In this section, we present our blockchain-based architecture for cooperative location privacy preservation in 5G-enabled vehicular fog computing. This section is structured as follows. We first describe the considered system model. We then present the system's initialization. Finally, we describe the attacker model. Table II presents the abbreviations and notations used throughout the paper.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2021.3083477, IEEE Transactions on Vehicular Technology

TABLE II: Abbreviations and notations used throughout the paper.

Notation	Description
PCP	Pseudonym Changing Process
BS	Base Station
CA	Certification authority
PCR	Pseudonym-Changing Requester
PCC	Pseudonym-Changing Cooperator
(PK_{bs_j}, $Cert_{bs_j}$)	bs_j's (public key, certificate)
($address_{v_i}$, $balance_{v_i}$)	v_i's (account address, balance)
(Rep_{v_i}, $K_{v_i,k}$)	v_i's (reputation, k^{th} pseudonym)
$Contract\_address$	the smart contract's address
ID_{pcr}	PCR's ID
(ID_{pcc_i}, π_{pcc_i})	PCC_i's (ID,payment)
C	PCP's price
σ	Penalty's price
$deposit_{pcr}$	PCR's deposit
$deposit_{pcc_i}$	PCC_i's deposit
CZ	Candidature Zone
$size_{CZ}$	CZ's size
SSC	Standard Smart Contract
OSC	Optimized Smart Contract
U_{bs_i}	bs_i's Utility value
$Score_{pcr_i}$	pcr_i's score
$Score_{pcc_j}$	pcc_j's score
max_{tp}	Maximum processing time
max_{tc}	Maximum consensus time
Ω	The set of consensus members

A. System Model

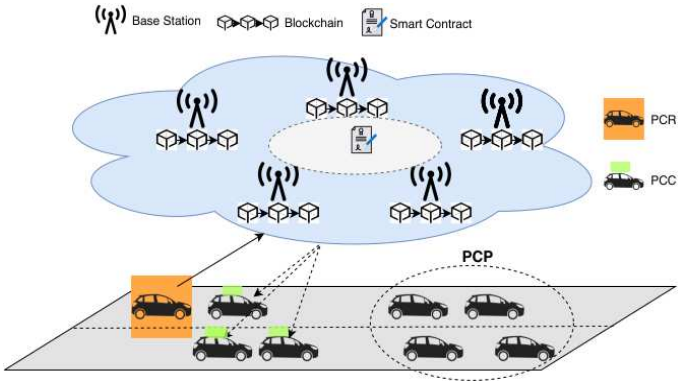

Fig. 1: Blockchain-based architecture for cooperative location privacy preservation in 5G-enabled vehicular fog computing

As illustrated in Figure 1, we consider a 5G-enabled vehicular fog computing architecture consisting of two layers. The infrastructure layer includes vehicles equipped with V2X technology. In this layer, communications are multi-hop Vehicle-to-Vehicle (V2V). Vehicle-to-Infrastructure (V2I) communications are only used to communicate with the 5G-fog layer. This latter consists of several Base Stations (BSs) acting as fog nodes with sufficient data storage, processing, and computing capabilities, and distributed over a specific geographic perimeter. All BSs are connected through secure 5G links. We also consider that each bs_j is equipped with a consortium blockchain hosting transactions and SCs for enabling secure cooperation between vehicles. Vehicles should carry out coordinated PCPs to protect their location privacy. They can then request PCPs from their neighbors. However, they cannot be sure that their neighbors will cooperate with them, which leads to the failure of PCPs. Consequently, vehicles may ask for support from our scheme. Indeed, 5G blockchain-based fog layer is acting as a controller of PCPs. All vehicles involved in these PCPs are protected by SCs while cooperation transactions are recorded in the consortium blockchain. In the following, we define a Pseudonym-Changing Requester (PCR) as each vehicle requests to perform a PCP i.e it requests to change its pseudonym with the neighboring vehicles. We also define a Pseudonym-Changing Cooperator (PCC) as each vehicle that participates in a PCP. The fog layer allows the rapid processing of the PCP's procedure from the PCR's request to the execution of the SC.

B. System Initialization

To implement an efficient cooperation between vehicles, before joining to the systems, vehicles and BSs register with the Certification Authority (CA). Specifically, during the registration, each bs_j is equipped with legitimate identity consisting of a private key SK_{bs_j}, a public key PK_{bs_j}, and a public certificate $Cert_{bs_j}$ respectively. On the other hand, each vehicle v_i is equipped with a legitimate identity consisting of a private key SK_{v_i}, a public key PK_{v_i}, and a public certificate $Cert_{v_i}$ respectively. Each vehicle v_i also gets an account $account_{v_i}$, which includes its wallet address $address_{v_i}$, its account balance $balance_{v_i}$, its reputation value Rep_{v_i}. Moreover, each vehicle v_i is pre-loaded with a set of s pseudonyms $K_{v_i,k}$ where k $\in 1, ..., s$, which are public keys certified by the CA. For each pseudonym $K_{v_i,k}$, the CA provides a certificate $Cert_{v_i,k}(K_{v_i,k})$. To ensure the authentication and integrity of information, asymmetric encryption is used in the architecture. Safety messages are properly signed with a private key $K^{-1}_{v_i,k}$ corresponding to the pseudonym $K_{v_i,k}$. A certificate is attached to each message to enable other vehicles to verify the sender's authenticity. In addition, each entity (vehicle/BS) is equipped with a security defense agent for thwarting internal attacks defined in subsection III-C. Each vehicle periodically broadcasts a safety message every t millisecond, where each message includes a location, a timestamp, a velocity, and a content. On the other hand, to maintain the privacy of vehicles in the blockchain, pseudonyms considered as the source address for verifying the authenticity of transactions. Pseudonyms are also used as account addresses. To this end, only CA still knows the relationship between the real identifier of the vehicle and its corresponding pseudonyms.

C. Attack Model

Malicious entities (vehicles/BSs) can have a significant impact on the scheme. In the following, we identify three types of attackers.

1) **Malicious PCR**: a malicious PCR can request to execute a PCP without having enough money in its balance or it can pretend that one or several PCC(s) didn't change its/their pseudonym(s) in the PCP.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2021.3083477, IEEE Transactions on Vehicular Technology

2) **Malicious PCC**: to be rewarded, a malicious PCC can pretend that it changes its pseudonym in PCP, but in reality it did not.
3) **Malicious BS**: a malicious BS tries to tamper PCP's related information such as reputations and data received from vehicles to increase its benefits.

Malicious entities can launch internal and denial of service attacks. They can also launch more advanced attacks like strategic attacks, where attackers disguise as PCCs first and then timely switch to malicious behaviors to threaten the proposed scheme.

IV. Cooperative Pseudonym Changing Smart Contract

In this section, we design a SC aiming at ensuring trust cooperation between vehicles and stimulating them to participate in PCPs. Each SC has a unique contract address ($Contract\_address$) and maintains a set of state variables including the identifier of the PCR (ID_{pcr}), the account address of the PCR ($account_{pcr}$), the identifiers of PCCs { ID_{pcc_1},..., ID_{pcc_n}}, the account addresses of PCCs {$account_{pcc_1}$,..., $account_{pcc_n}$}, the price of the PCP C .i.e the total number of coins the PCR that pays for the PCCs. The contract also includes the number of coins to pay for each PCC {π_{pcc_1},..., π_{pcc_n}}, the penalty price (σ) applied to a PCC if case of non cooperation, ρ is the service ratio to calculate the number of coins to pay network operators managing BSs from C, the time when a PCR requests the creation of the smart contrat ($trequest$), the time when the SC is effectively created ($creation\_time$), the time when the PCP is performed (t_{pcp}), and the closing time of the SC ($close\_time$). In addition, to protect against malicious PCR and PCCs, the PCR and each PCC should move a deposit from their wallet addresses to the contract address. Specifically, the PCR and PCCs should move numbers of coins to ($deposit_{pcr}$) and {$deposit_{pcc_1}$,..., $deposit_{pcc_n}$} respectively. Algorithm 1 describes the implementation of the SC. The pseudonym changing SC consists of one public function, which can be called by vehicles, and four private functions, which can only locally be called by the BS.

Algorithm 1: Cooperative Pseudonym Changing Smart Contract

1 **State variables:**;
2 $Contract\_address$, ID_{pcr}, {ID_{pcc_1},..., ID_{pcc_n}};
3 $account_{pcr}$, {$account_{pcc_1}$..., $account_{pcc_n}$};
4 C, σ, {π_{pcc_1}, π_{pcc_2},..., π_{pcc_n}};
5 {$deposit_{pcc_1}$, $deposit_{pcc_2}$,..., $deposit_{pcc_n}$} $trequest$, $deposit_{pcr}$, t_{pcp}, $creation\_time$ $close\_time$;
7 **public Create()**
8 ***Input:*** $Req^{pcr \to bs_j}$;
9 **if** *(*$Rep_{pcr} > 0$*) and (*$balance_{pcr} >= c(1+\rho)$ *)* **then**
10 $Contract\_address \leftarrow H(ts \mid\mid K_{pcr})$;
11 $ID_{pcr} \leftarrow K_{pcr}$; $account_{pcr} \leftarrow addr_{pcr}$;
12 $deposit_{pcr} \leftarrow$ **Move**($balance_{pcr}$, c);
13 $C \leftarrow c$; $\sigma \leftarrow c$; $trequest \leftarrow ts$;
14 **else**
15 $Rep_{pcr} \leftarrow (Rep_{pcr}$ - 1) ; $Consensus()$;
16 **end**
18 **private Negotiate()**
19 ***Input:*** { $Mes^{v_1 \to bs_j}$,..., $Mes^{v_m \to bs_j}$ } ;
20 {v_1,..., v_l} $\leftarrow$ **Match**({loc_{v_1},... , loc_{v_m}}, loc_{pcr}, $size_{cz}$) ;
21 {π_{v_1},...,π_{v_l}} $\leftarrow$ formula (1);
22 $\forall\ v_i \in \{v_1,..., v_l\}$: **Send** ($Mes^{bs_j \to v_i}(\pi_{v_i}, \sigma)$) ;
24 **private Deploy()**
25 ***Input:*** {$Resp_{msg}^{v_1 \to bs_j}$,..., $Resp_{msg}^{v_l \to bs_j}$ } ;
26 {pcc_1,..., $pcc_{n'}$}, {v_1,..., $v_{l-n'}$} $\leftarrow$ **Analyze** ({$Resp_{msg}^{v_1 \to bs_j}$,..., $Resp_{msg}^{v_l \to bs_j}$ });
27 **for** $v_i \in \{pcc_1,..., pcc_{n'}\}$ **do**
28 **if** *(*$balance_{v_i} > 0$*)* **then**
29 $ID_{pcc_i} \leftarrow K_{v_i}$; $account_{pcc_i} \leftarrow addr_{v_i}$;
30 $\pi_{pcc_i} \leftarrow$ formula (1);
31 **if** *(*$balance_{v_i} >= \sigma$ *)* **then**
32 $deposit_{pcc_i} \leftarrow$ **Move**($balance_{pcc_i}$, σ);
33 $Rep_{v_i} \leftarrow (Rep_{v_i}$ + 1);
34 **else**
35 $deposit_{pcc_i} \leftarrow$ **Move**($balance_{pcc_i}$);
36 $Rep_{v_i} \leftarrow (Rep_{v_i}$ + 0.5);
37 **end**
38 **end**
39 **end**
40 $\forall\ v_i \in \{v_1,..., v_{l-n'}\}$: $Rep_{v_i} \leftarrow (Rep_{v_i}$ - 1);
41 **if** ***Consensus()***==*true* **then**
42 $creation\_time \leftarrow timestamp$; set(t_{pcp});
43 **Send** ($Conf^{b_j \to pcr}(t_{pcc})$);
44 $\forall\ v_i \in \{pcc_1,..., pcc_n\}$: **Send** ($Conf^{b_j \to v_i}(t_{pcc})$);
45 **end**
47 **private Invoke()**
48 ***Input:*** $Fb^{pcr \to bs_j}$, {$Fb^{pcc_1 \to bs_j}$,..., $Fb^{pcc_n \to bs_j}$} ;
49 $Execute\_contract()$; $Close()$;
51 **private close()**
52 $close\_time \leftarrow timestamp$; $Consensus()$;

A. Create

When a vehicle v_i (PCR) wants to execute a PCP, it needs to call the create function. Thus, it sends a request to the nearest bs_j: $Req^{pcr \to bs_j} = EPK_{bs_j}(addr_{pcr}||c||loc_{pcr}||K_{pcr}||Sig_{K_{pcr}}||Cert_{K_{pcr}}||ts)$. This request is encrypted by PK_{bs_j} and includes the PCR's account address ($addr_{pcr}$), the price to pay to perform this operation (c), the current location (loc_{pcr}), PCR's current pseudonym (K_{pcr}), the corresponding signature ($Sig_{(K_{pcr})}$), certificate ($Cert_{K_{pcr}}$), and a timestamp ts. Once bs_j receives a request from a PCR, it first checks Rep_{pcr} and $balance_{pcr}$ to verify if its reputation is positive and it has enough coins to pay for PCCs and service fees in step (9). If the condition is satisfied, the SC is created and a unique identifier is assigned to the contract address in step (10), which consists of the hash value of the concatenation of the timestamp and the current pseudonym of the PCR. The state variables (ID_{pcr} and $account_{pcr}$) related to PCR are also initialized in steps (11) and a deposit of c coins is moved from $balance_{pcr}$ to $deposit_{pcr}$ in step (12). C, σ, and $trequest$ are also initialized in step (13). However, if the PCR tries to execute a PCP without having enough coins in its balance, the request is refused and the reputation value of the PCR is decreased in step (15). A consensus process should also be done latter to update the blockchain ledger.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2021.3083477, IEEE Transactions on Vehicular Technology

B. Negotiate

After creating the SC, a set of PCCs should be selected to participate with the PCR in the next PCP. For this reason, each bs_j keeps monitoring vehicles under its coverage. Since the privacy level is not part of the standard structure of the beacon, each vehicle (v_i) then periodically broadcasts a message to bs_j: $Mes^{v_i \rightarrow bs_j} = EPK_{bs_j}(loc_{v_i}||pv_{v_i}||K_{v_i}||Sig_{K_{v_i}}||Cert_{K_{v_i}}||ts)$. This message is encrypted by PK_{bs_j} and includes the current position of the vehicle (loc_{v_i}) and its privacy level (pv_{v_i}). It also includes v_i's current pseudonym (K_{v_i}), the corresponding signature ($Sig_{K_{v_i}}$), certificate ($Cert_{K_{v_i}}$), and timestamp (ts). This message should be encrypted since the privacy level is private information. Sharing this information can have social impacts on drivers.

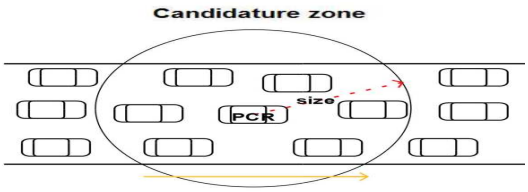

Fig. 2: Candidature zone for a given PCR

In step (20), once a PCR's request is received by bs_j, it matches between the received request and the monitoring (m) vehicles to select the l vehicles {v_1,..., v_l} from the candidature zone of the PCR. As illustrated in Figure 2, the Candidature Zone (CZ) is defined as the road area that contains the potential candidate vehicles that can cooperate with the PCR in its PCP. More specifically, CZ is a circle whose center is the position of the vehicle and its radius is the size of CZ, denoted as $size_{CZ}$. After selecting the potential cooperative candidates, bs_j calculates the number of coins to pay for each candidate vehicle {π_{v_1},..., π_{v_l}}. The need to participate in the PCP is different from a vehicle to another according to its current privacy level. In addition, the reputation value of a vehicle is a good indicator of the level of cooperation of vehicles. To this end, we adopt the payment of cooperative vehicles according to their privacy levels and their reputation values. In other words, vehicles with high privacy levels and reputation values will be paid more for rewarding them for their cooperative behavior and for their sacrifices since their need to change their pseudonyms is weak compared to other vehicles. In step (21), the payments of vehicles are calculated according to their privacy levels and reputation values using the following formula:

$$\pi_{v_i} = \frac{rep_{v_i} * pv_{v_i}}{\sum_{j=1}^{l}(rep_{v_j} * pv_{v_j})} * C \quad (1)$$

In step (22), once of the calculation of the payments of cooperative vehicles is done, bs_j sends a message for each selected vehicle: $Mes^{bs_j \rightarrow v_i} = EK_{v_i}(\pi_{v_i}||\sigma||Sig_{PK_{bs_j}}||ts)$. This messages is encrypted by the current vehicle's pseudonym (K_{v_i}) and includes the number of coins (π_{v_i}), which the vehicle will receive in case of cooperation, the penalty price (σ) applied to the vehicle in case of no respect of SC's clauses, and the signature ($Sig_{PK_{bs_j}}$) and the timestamp ts.

C. Deploy

Before deploying the SC into the consortium blockchain, bs_j needs to wait for responses from candidates vehicles to check their willingness to participate in the PCP: $Resp_{msg}^{v_i \rightarrow bs_j} = EPK_{bs_j}(resp_{v_i}||addr_{v_i}||K_{v_i}||Sig_{K_{v_i}}||Cert_{K_{v_i}}||ts)$. These responses are encrypted by PK_{bs_j} and include the cooperation decision of the candidate vehicle ($resp_{v_i}$), (v_i)'account address ($addr_{v_i}$), v_i's current pseudonym (K_{v_i}), the corresponding signature ($Sig_{K_{v_i}}$), certificate ($Cert_{K_{v_i}}$), and timestamp (ts).

In step (26), the response messages are analyzed to distinguish between cooperative vehicles and non-cooperative vehicles. The balance of each cooperative vehicle is checked in step (28). If the balance is positive, the vehicle assigned as an PCC and its related parameters (ID_{pcc_i}, $account_{pcc_i}$) are initialized in step (29). In step (30), a recalculation of the vehicle's payment using the formula 1 is also necessary since the number of selected l vehicles may differ from the number of cooperative vehicles. In addition, in step (31), b_j checks if the vehicle has enough coins to pay for the penalty σ (if applicable). If the check passes, then a deposit of σ coins is moved from the vehicle's balance to the contract address in step (32) and the vehicle's reputation is increased by 1 in step (33). Otherwise, existing coins in the vehicle's balance are moved to the contract address in step (35) but the vehicle's reputation is increased by only 0.5 in step (35). On the other hand, the reputations values of all non-cooperative vehicles ($l - n'$) will be decreased by 1 in step (40).

In this stage, the SC is ready to be deployed into the blockchain. After reaching consensus in the consortium blockchain, the SC is successfully deployed and can be accessed by all the blockchain nodes. Once the contract is deployed, bs_j sets $creation\_time$ and t_{pcp} in step (42). Then, it sends a confirmation message to the PCR : $Conf^{bs_j \rightarrow pcr} = EK_{pcr_i}(Contract\_address||t_{pcp}||Sig_{PK_{bs_j}}||ts)$ in step (43). A confirmation is also sent to each PCC {pcc_1,..., pcc_n} in step (44): $Conf^{bs_j \rightarrow pcc_i} = EK_{pcc_i}(Contract\_address||t_{pcp}||\pi_{pcc_i}||Sig_{PK_{bs_j}}||ts)$. The confirmations message include the contract address ($Contract\_address$), (t_{pcp}), the signature $Sig_{PK_{bs_j}}$ and the timestamp ts. In addition, $Conf^{bs_j \rightarrow pcc_i}$ includes the amount of coins should each PCC gets after having participated in the PCP.

D. Invoke

This function is automatically called by bs_j as soon as ($t >= t_{pcp}$) to perform necessary transactions and financial settlements. This function needs an input from the PCR and each PCC to verify whether PCP is executed according to the SC clauses. Necessary penalties followed by decreasing reputation values are also applied to malicious PCCs. Specifically, PCR sends a feedback message to bs_j: $Fb^{pcr \rightarrow bs_j} = EPK_{bs_j}(Contract\_address||${$pcc_1$,...,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2021.3083477, IEEE Transactions on Vehicular Technology

$pcc_n\}||K_{pcr}||Sig_{K_{pcr}}||Cert_{K_{pcr}}||ts)$. This message is encrypted by PK_{bs_j} and includes the contract address ($Contract\_address$), the pseudonyms of vehicles that change their pseudonyms in the PCR's PCP. This message also includes PCR's current pseudonym (K_{pcr}), the corresponding signature ($Sig_{K_{pcr}}$), certificate ($Cert_{K_{pcr}}$), and a timestamp (ts). Each PCC should also send a feedback message to bs_j to confirm its participation in the PCP: $Fb^{pcc_i \rightarrow bs_j} = EPK_{bs_j}(Contract\_address||K_{pcc_i}||K^{'}_{pcc_i}||Sig_{K_{pcc_i}}||Cert_{K_{pcc_i}}||ts)$. This message is also encrypted (PK_{bs_j}) and includes the contract address ($Contract\_address$), the PCR's current pseudonym (K_{pcc_i}), the PCR's previous pseudonym ($K^{'}_{pcc_i}$), the corresponding signature $Sig_{K_{pcc_i}}$, certificate $Cert_{K_{pcc_i}}$, and timestamp ts. Once bs_j receives these confirmation messages, it executes the SC is in step (49). Thus, the financial transactions concerning payments and penalties are generated and prepared for block building. Finally, the function $Close()$ is called for running the consensus progress and closing the smart contact.

E. Close

This function starts by deactivating all the functions of the SC and assigning the close time ($close\_time$). Then, a consensus process is executed in step (52) to update the ledger, as described in Section VI.

V. Smart Contract Optimization

In this section, we propose an optimization for the pseudonym cooperation SC. The goals of this optimization are to (i) minimize the number of smart contracts managed by the scheme, (ii) reduce the price paid by PCRs, and (iii) increase the location privacy levels obtained in PCPs. To implement the SC optimization process, during ΔT_1, bs_j collects requests for PCRs. At the end of this period, bs_j runs a k-means algorithm [36] to group PCRs into clusters according to their positions and their directions. PCRs within the same cluster will participate in the same PCP. In the following, we denote the SC described in the previous section IV as the Standard SC (SSC). The Optimized SC (OSC) is derived from the SSC and its implementation is given in Algorithm 2. Unlike the SSC, which is one-to-many SC between one PCR and multiple PCCs, the OSC is a many-to-many SC between multiple PCRs and multiple PCRs. Thus, in the OSC, the state variables ID_{pcr}, $account_{pcr}$, $deposit_{pcr}$ are replaced by $\{ID_{pcr_0},...,ID_{pcr_n}\}$, $\{account_{pcr_0},...,account_{pcr_n}\}$, $\{deposit_{pcr_0},...,deposit_{pcr_n}\}$ respectively. The OSC contains the same functions as the SSC, but all of them are private. In the following, we present the main optimizations in these functions:

A. Create

Unlike the SSC, the function $create$ turns to private in the OSC. As aforementioned, an OSC is created for each cluster of PCRs. The $create$ function then takes the group of requests belonging to the same cluster as an input. For each request $Req^{pcr_i \rightarrow bs_j}(c_i)$, bs_j checks if the vehicle has a positive reputation and enough coins to pay for the PCP. If this condition is satisfied, the vehicle will be assigned as a pcr_i and its related state variables (ID_{pcr_i}, $account_{pcr_i}$) will be initialized in steps (9) and (10) respectively. In addition, in step (11), a number of coins (c_i) is moved from the PCR's balance to the contract address as a deposit ($desposit_{pcr_i}$), and, in step (12), $trequest$ is initialized. However, in the case of the vehicle's balance is less than c_i, its reputation value is decreased in step (15). Thus, a consensus process is necessary later in step (18) to keep the values of reputation updated in the ledger. Furthermore, in step (19), the contract address is initialized to the hash of the concatenated pseudonyms of PCRs and the timestamp. Also, in step (20), the total number of coins to pay (costs) for the PCP is initialized by the average number of coins offered by PCRs, which is calculated using the following formula:

$$C = \frac{1}{n}\sum_{j=1}^{n} c_j \tag{2}$$

Since the reputation values of PCRs are different, we propose to adapt the contribution of each PCR in the total costs (C) according to its reputation in step 21. In other words, PCRs with high reputation values will pay less more than other vehicles. The contribution of each of PCR is computing using the following formula:

$$contrib_i = \frac{C}{Rep_{pcr_i} * \sum_{j=1}^{n} \frac{1}{Rep_{pcr_j}}} \tag{3}$$

B. Negotiate

In step 26, unlike the SSC, the OSC matches between the positions of monitoring vehicles and PCRs' positions of the same cluster to select the candidate vehicles. Therefore, as shown in Figure 3, the CZ of the OSC is the union of CZs of these PCRs.

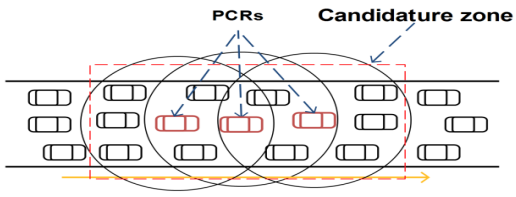

Fig. 3: Candidature zone for the optimized smart contract.

C. Deploy

The OSC executes the same code as the SSC. Except, in step 36, once the contract is deployed, bs_j sends a confirmation to each PCR specifying the $contract\_address$, t_{pcc} and its contribution to the total costs of the PCP ($contrib_{pcr}$), which is calculated using the formula 3.

D. Invoke

Compared to the SSC, the OSC takes feedback for each PCR participating in the PCP.

Algorithm 2: Optimized Smart contract implementation algorithm

State variables:;
  $\{ID_{pcr_1}$,...,$ID_{pcr_n}\}$, $\{account_{pcr_1}$,...,$account_{pcr_n}\}$, $\{deposit_{pcr_1}$,...,$deposit_{pcr_n}\}$, $\{contrib_{pcr_1}$,...,$contrib_{pcr_n}\}$;
@*Override*
private Create()
  Input: group({ $Req^{pcr_1 \rightarrow bs_j}(c_1)$,..., $Req^{pcr_n \rightarrow bs_j}(c_n)$}) ;
  for $i \in \{0,..., n\}$ **do**
    if ($Rep_{pcr_i} > 0$) *and* ($balance_{pcr_i} >= c$) **then**
      $ID_{pcr_i} \leftarrow K_{pcr_i}$;
      $account_{pcr_i} \leftarrow addr_{pcr_i}$;
      $deposit_{pcr_i} \leftarrow$ **Move**($balance_{pcr_i}$, c_i);
      $trequest \leftarrow ts$;
      $ca \leftarrow ca \mid\mid K_{pcr_i}$;
    else
      $Rep_{pcr_i} \leftarrow (Rep_{pcr_i}$ - 1);
    end
  end
  $Consensus()$;
  $Contract\_address \leftarrow H(ts \mid\mid ca)$;
  $C \leftarrow$ formula (2); $\sigma \leftarrow$ formula (2) ;
  $contrib_{pcr_i} \leftarrow$ formula (3);
@*Override*
private Negotiate()
  Input: { $Mes^{v_1 \rightarrow bs_j}$,..., $Mes^{v_m \rightarrow bs_j}$ } ;
  $\{v_1$,..., $v_l\} \leftarrow$ **Match**($\{loc_{v_1}$,... , $loc_{v_m}\}$, { loc_{pcr_i},...,$loc_{pcr_i}\}$, $size_{cz}$);
  $\{\pi_{v_1}$,...,$\pi_{v_l}\} \leftarrow$ **Calculate_pay** ($\{pv_{v_i}$,..., $pv_{v_l}\}$);
  $\forall\ v_i \in \{v_1$,..., $v_l\}$: **Send** ($Mes^{b_j \rightarrow v_i}(\pi_{v_i}, \sigma)$) ;
@*Override*
private Deploy()
  Input: $\{Resp_{msg}^{v_1 \rightarrow bs_j}$,..., $Resp_{msg}^{v_l \rightarrow bs_j}$ } ;
  Super.Deploy();
  if ***Consensus()*****==***true*** then**
    set(t_{pcc});
    $\forall\ v_i \in \{pcr_1$,..., $pcr_{n1}\}$:
      Send ($Conf^{bs_j \rightarrow pcr}$ (t_{pcc_i}, $contrib_i$));
    $\forall\ v_i \in \{pcc_1$,..., $pcc_{n2}\}$:
      Send ($Conf^{bs_j \rightarrow v_i}(t_{pcc})$);
    $creation\_time \leftarrow timestamp$;
  end
@*Override*
private Invoke()
  Input: $\{Fb^{pcr_1 \rightarrow bs_j}$,...,$Fb^{pcr_{n1} \rightarrow bs_j}\}$, $\{Fb^{pcc_1 \rightarrow bs_j}$,..., $Fb^{pcc_{n2} \rightarrow bs_j}\}$;
  Super.Invoke();
@*Override*
private close()
  Super.Close();

E. Close

No change is done in this function compared to the SSC.

VI. Utility-based Delegated Byzantine Fault Tolerance Consensus Protocol

Consensus processes should be carried to ensure that each member of the consortium blockchain has a coherent and recognized of the whole ledger. To efficiently reach the consensus in our scheme, we propose a Utility-based Delegated Byzantine Fault Tolerance (U-DBFT) consensus protocol, which is based on [37]. The consensus protocol comprises two steps: (i) the consensus members and leader selection, and (ii) the consensus process.

A. Consensus members and leader selection

The members of the consortium blockchain (BSs) have two types of roles: simple members and consensus members. While a consensus member can participate in consensus processes, a simple member can only broadcast transactions into the blockchain network and accept the validated blocks.

The selection of the consensus members is done according to their utility values $\{U_{bs_1}, U_{bs_2}, ..., U_{bs_3}, ...U_{bs_n}\}$, which are calculated on the basis scores received from vehicles. As shown in formula 4, the top (Ω) BSs with the highest utility values are selected as consensus members. The set of consensus members is denoted as Ω = {1,...,Ω}. We assume that $\Omega >= 3f$ +1, where f is the maximum number of malicious members in the consortium blockchain.

$$\{bs_1, ...bs_\Omega\} = Max(\{U_{bs_1}, U_{bs_2}, ..., U_{bs_3}, ...U_{bs_n}\}, \Omega) \quad (4)$$

As shown in formula 5, U_{bs_k}, the utility of a bs_k is the average of the sum of scores calculated by PCR(s) and PCCs weighted by their reputation values. It is worth mentioning that depending on a PCP, a vehicle can be either a PCR or a PCC. In addition, during their journey on the road, vehicles can participate in several PCPs. Thus, before being out of the coverage of bs_k, each vehicle sends its scoring values to bs_j.

$$\begin{aligned} U_{bs_k} = \frac{1}{nb_{sco_{pcr}}} * \sum_{i=1}^{nb_{sco_{pcr}}} (Rep_{pcr_i} * Score_{pcr_i}) \\ + \frac{1}{nb_{sco_{pcc}}} * \sum_{j=1}^{nb_{sco_{pcc}}} (Rep_{pcc_j} * Score_{pcc_j}) \end{aligned} \quad (5)$$

$nb_{sco_{pcr}}$ and $nb_{sco_{pcc}}$ are the number of scores received from PCRs and PCCs respectively. $Score_{pcr_i}$ is the score given by a pcr_i to bs_k, which is calculated using formula 6. In this formula, $nb1_{pcp}$ is the number of PCPs where the vehicle is involved as a PCR. The score of a bs_k is calculated based on the average of the sum of values obtained in each executed PCP. These values are the value to money (vm) given by the formula 8, which assesses the monetary cost against the location privacy obtained after executing a PCP, the processing speed (ps) given by the formula 9, which assesses the speed of establishing the SC, and the consensus speed of the block (cs), given by the formula 10.

$$Score_{pcr} = \frac{\sum_{i=1}^{nb1_{pcp}} (vm_i + cs_i + ps_i)}{nb_{pcp}} \quad (6)$$

On the other hand, $Score_{pcc}$ is the score given by a pcc_i to bs_k. As shown in the formula 7, a pcc calculates the score according to the average of the sum of cs (formula 9) and ps (formula 10) values obtained after each PCP. Here, $nb2_{pcp}$ is the number of PCPs where the vehicle is involved as a PCC. Moreover, this score also takes into the account the number of PCPs' proposals (nb_{prop}) received from the bs_k.

$$Score_{pcc} = \frac{\sum_{i=1}^{nb_{pcp}} (cs_i + ps_i)}{nb_{pcp}} + nb_{prop} \quad (7)$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2021.3083477, IEEE Transactions on Vehicular Technology

As aforementioned, vm is a metric to assess the monetary cost against the location privacy level obtained executing a PCP. vm is calculated using the following formula, where $priv$ is the obtained location privacy level, and $pric$ is the price paid for a given PCP.

$$vm = \frac{priv}{pric} \tag{8}$$

ps is a metric to assess the effort taken by a bs to establish a SC, which includes the selection of the candidate vehicles and sending/receiving messages. ps is calculated using the following formula, where t_p is the effective processing time, and max_{tp} is the maximum expected time for processing.

$$ps = \frac{max_{tp} - tp}{max_{tp}} \tag{9}$$

cs is the consensus speed, which is a metric defined to measures how faster the consensus process was done from the block production to the block insertion in the consortium blockchain. cs is calculated using the following formula where t_c is the effective time for the consensus process and max_{tc} is the maximum taken for the consensus process of one block.

$$cs = \frac{max_{tc} - tc}{max_{tc}} \tag{10}$$

A network operator who manages a set of BSs aims that their BSs are part of the set of the consensus members to receive coins for each performed consensus process. Thus, BSs will do their best to increase their utility values for participating in the consensus process. However, BSs will try to tamper the score values of vehicles for increasing their utility and thereby monopolizing the consensus process. For this reason, we also propose to store the utility values into the ledger. The scores of vehicles are broadcast to the blockchain. Each ΔT_2, the utility values of BSs are calculated and a consensus process is carried out to update the set of the consensus members (Ω).

In our scheme, the first leader is the member with the highest utility value. After that, the leader p is changed after a each consensus process or if it fails during the current consensus process. The selection of the next leader p is done according to a round-robin (circular) policy using the following formula:

$$p = v \ mod \ \Omega \tag{11}$$

In Ω, the consensus members are in descending order according to their utility values starting from index 0. Based on [37] a view v is a period of time in which a given consortium member is the leader. In formula 11, v is an identifier of a given period of time. Therefore, a view change means switching to a different leader.

B. Consensus process

The consensus process runs by a consensus member (i) is described in Algorithm 3. This algorithm describes the consensus process applied to the blocks (the transactions and states) related to a (SC). However, the same consensus process is applied to the blocks related to the reputation and utility values. Here t_s is a transaction record related to the SC, $\Upsilon_{i,s}$ is a set of SC's transactions validated by the consensus member i. $\Phi_{i,s}$ is the updated state after the execution of SC with the set of corresponded transactions Υ_s. B_i is a local block created by the consensus member i, $verify\_transaction(t_s)$ is a function to the verify the validity of a transaction t_s, $execute(\Upsilon_{i,s})$ is a function that locally executes the SC with the corresponded transactions $\Upsilon_{i,s}$, BuildBlock ($\Upsilon_{i,s}$, $\Phi_{i,s}$) is to build local block with the transaction set $\Upsilon_{i,s}$ and the state set $\Phi_{i,s}$. The consensus process then contains the following steps:

Algorithm 3: Utility-based DBFT Consensus Protocol

```
v ← 0, k ← 1 ;
Broadcast()
    Input: transaction t_x;
    broadcast(t_x);
Collect()
    Input: transaction t_s;
    if i ∈ Ω then
        if (verify_transaction (t_s) == true) then
            Υ_{i,s} ← Υ_{i,s} ∪ t_s;
        end
        if (all transactions of the contract are received) then
            Φ_{i,s} ← execute(s,Υ_{i,s});
            B_{i,s} ← BuildBlock (Υ_{i,s}, Φ_{i,s});
        end
    end
Propose()
    Input: block B_{i,s} ;
    if leader(i) ==true then
        broadcast (proposal, i, v, B_{i,s}, Sig_{SK_{bs_i}}(H(B_{i,s}));
    end
Confirm()
    Input: given block B_{j,s};
    if i ∈ Ω and leader(i) == false then
        if VerifyBlock(B_{j,s}) == true and getState(B_{j,s}) ==
          Φ_{i,s} then
            Broadcast(Confirm,i, v, Sig_{SK_{bs_i}}(H(B_{j,s}));
        else
            k ← k + 1; v_k ← v + k;
            Broadcast(Changeview,i, v, v_k);
        end
    end
Publish
    Input: Message msg;
    if Confirmation(msg) == true then
        ConfirmMsg ++;
    end
    if ChangeView(msg,v_k) == true then
        ChgMsg ++;
    end
    if (ConfirmMsg >= Ω − f) then
        PublishBlock();
        k ← k + 1 ; v_k ← v + k ;
        SelectNewLeader() by using formula 11; ;
    end
    if (t >= max_tc or (ChgMsg >= Ω − f) then
        SelectNewLeader() by using formula 11 ;
        StartNextRound();
    end
```

1) Broadcast: When an SC is triggered between PCR(s) and PCCs under the coverage of bs_j, this latter broadcasts all the corresponded transactions into the whole consortium blockchain for audit and verification.

2) Collect: All consensus members collect all SC's transactions. Each transaction t_s is verified in step (10) and only the validated transactions are added to the list of validated transactions $\Upsilon_{i,s}$ in step (11). Each consensus member waits to receive all the SC's transactions before it locally executes the SC in step (14). The changed states after executing the SC are saved in the local state ledger of each consensus member. All validated transactions and states are ordered by the timestamp and packaged into a block in step (15). Building a local block by each consensus member significantly reduces the time of verifying candidate blocks. Indeed, a no-leader consensus member can verify a candidate block received from the leader by simply comparing its local block with the candidate block.

3) Propose: After all non-leader consensus members have finished building their local blocks, the leader consensus member broadcasts a proposal to all non-leader consensus members in step (22). This proposal includes leader's identifier (i), the view v, the local block $(B_{i,s})$, and the hash value of the block $(H(B_{i,s}))$ singed by SK_{bs_i}.

4) Confirm: Once a non-leader vehicle receives a candidate block $B_{j,s}$, it first verifies its validity using $verifyBlock()$, then it uses the function $getState()$ to retrieve the state of the block for comparing it with its local state $\Phi_{i,s}$. If these checks passed, each non-leader consensus member broadcasts a confirmation message in step (29), which includes its identifier i, the view change v and the signature the hash of the block $(Sig_{SK_{bs_i}}(H(B_{j,s}))$. However, if the received block is not valid, the view change will be triggered, where the next view change v_k is calculated in step (31). Therefore, the non-leader consensus member will broadcast the $changeviewMsg$ message in step (32), which includes non-leader's identifier (i), the current view v, and the changed view v_k.

5) Publish: Each consensus member keeps counting the number of received confirmations and the number of views changes v_k in steps (39) and (42) respectively. If the number of received confirmation messages is no less $(\omega - f)$ massages from other distinct consensus members, the consensus is reached and the block is ready to be published in the blockchain. To ensure the tractability and verification, each block is added in a chronological order into the blockchain and includes a cryptographic hash to the prior block. To prepare for the next consensus process, the view is changed in step (46) and the next leader is selected in step (47) using the formula 11. However, if the max period to reach the consensus (max_{tc}) has passed or the number of received view change messages with the same v_k is at least $(\Omega - f)$ from distinct consensus members, a new leader is selected in step (50) and the next round of the consensus process will start in step (51).

VII. Performance Evaluation

In this section, we evaluate the performance of the proposed scheme. We first evaluate the cooperative behavior in our scheme. We then perform a monetary analysis on the payments received by PCCs and the costs paid by PCRs considering both SSCs and OSCs. In addition, we evaluate the time needed to reach the consensus in the consortium blockchain and carry out an analytic evaluation of the utility function used to select the first leader in the set of the consensus members. Finally, we formulate a security game to capture different attacker behaviors in our scheme.

A. Cooperative behavior

We have carried out a set of simulations to evaluate the cooperative behavior of vehicles in our scheme. We first study the average number of cooperative vehicles inside PCPs in our scheme compared to random and basic cooperation strategies. We then evaluate the impact of variating both traffic density (ρ) and the size of the candidature zone ($size_{CZ}$) on the average number of cooperative vehicles inside PCPs and the number of performed PCPs respectively. Finally, we compare the number of created SCs and the average number of vehicles per SC considering both SSCs and OSCs.

TABLE III: Simulation Parameters

Parameter	**Value**
Simulation duration	60 s
Transmission Range	500 m
Mobility Model	krauß
Traffic density	{60,80,100,120,140} veh/km
Initial privacy levels	$\mathcal{N}$ ($\mu = 12$, $\sigma = 1.33$)
Initial reputation values	$[0.1, 1]$
Sensitivity parameters	$\mathcal{N}$ ($\mu = 0.1$, $\sigma = 0.011$)
Privacy threshold	5
$size_{CZ}$	{30, 60, 90} m
C	100 coins

These simulations are conducted using Veins simulation Framework [38]. We considered the case of a freeway road. We simulated a 3-lane straight road section of 3 Km. The mobility of vehicles is generated using SUMO [39] and follows the krauß mobility model [40]. As shown in Table III, we consider that traffic density is ranging from 60 to 140 vehicles/km. The initial reputation values of vehicles are randomly initialized with values $\in [0.1, 1]$. The privacy level values of vehicles are initialized according to a normal distribution $\mathcal{N}$ ($\mu = 12$, $\sigma = 1.33$). To capture the location privacy level as a function of the power of the adversary, we adopt the user-centric model proposed in [13]. The loss of location privacy of vehicles is modeled using a linear function, where the privacy loss increases with time according to a sensitivity parameter, $0 < \lambda_i < 1$. This maximum value of privacy loss is the location privacy protection level achieved at the last PCP. The loss of privacy is set to 0 after each PCP. In our simulations, we consider that sensitivity values of vehicles are initialized according to a normal distribution $\mathcal{N}$ ($\mu = 0.1$, $\sigma = 0.011$). Vehicles look to perform PCPs when their privacy levels are close to the privacy threshold, which is set to 5. We also consider different values of $size_{CZ}$. We run simulation several times calculate the average value of 95% confidence interval.

Figure 4 compares the average number of cooperative vehicles inside PCPs in our scheme with two cooperative strategies: random and basic. The random strategy represents a naive cooperative behavior, where vehicles take the cooperation decision without considering their self-interests. In the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2021.3083477, IEEE Transactions on Vehicular Technology

basic cooperative strategy, vehicles participate in the PCP only if their privacy levels go blow the privacy threshold. The results show that the average number of cooperative vehicles in our scheme is higher than the random and basic strategies whatever the traffic density is.

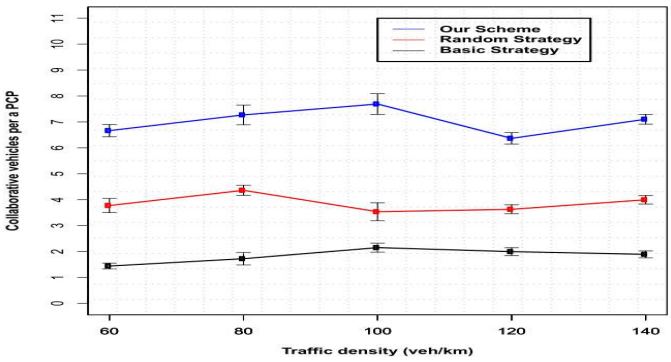

Fig. 4: The average number of cooperative vehicles per PCP as a function of traffic density comparing our scheme with two cooperation strategies ($size_{CZ}$ = 60 m)

In Figure 5, we evaluate the impact of variating the $size_{CZ}$ on the average number of cooperative vehicles per PCP over different traffic densities. Our results show that the number of cooperative vehicles increases with $size_{CZ}$. However, numbers remain stable over different traffic density levels. This is mainly due to the predefined parameters of the mobility model such as the safety distance and changing lane strategies, which prevents having more vehicles in CZs when the traffic density increases. This leads to an increase in the number of performed PCPs with the increase of the traffic density, as we can see in Figure 6. Indeed, the smaller the CZs, the faster the number of the performed PCPs increases with the traffic density.

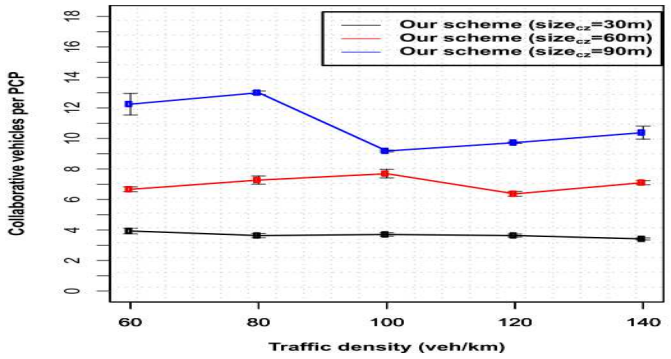

Fig. 5: The average number of cooperative vehicles per PCP as a function of traffic density variating $size_{CZ}$

In our previous evaluations, we consider that PCPs only run under SSCs. In the following evaluation, we compare two scenarios: (i) PCPs running under SSCs, and (ii) PCPs running under OSCs. The scikit-learn python library (*https://scikit-learn.org*) is used to run k-means clustering with $k = 4$ to create groups of PCRs associated with OSCs. Figure 7 (a) shows the number of PCRs per each OSC. Figure 7 (b) compares the number of SCs created in each scenario. The

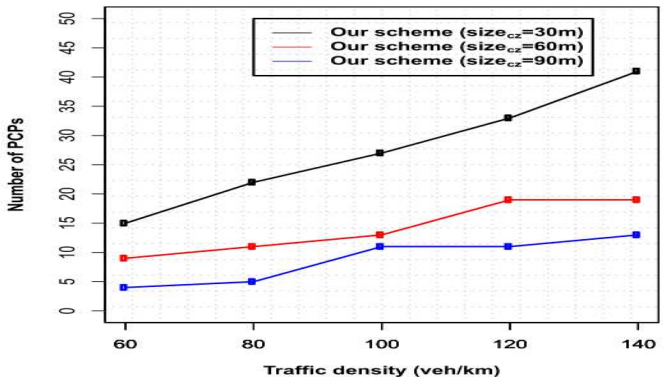

Fig. 6: The number of performed PCPs as a function of traffic density variating $size_{CZ}$

results show that using OSCs, our scheme can save more 65% of the total number of SCs. In addition, Figure7 (c) shows that the average number of cooperative vehicles inside PCPs is higher when using OSCs. These results confirm that OSCs allows reducing the number of SCs managed by the scheme and increase the privacy level obtained in PCPs.

B. Monetary analysis

In this section, we perform monetary analysis of payments received by PCPs and the cost paid by PCRs under SSCs and OSCs. Figure 9 shows the payments received by five PCCs in three different PCPs.

TABLE IV: The privacy levels and reputation values of five vehicles in three different PCPs

		PCC1	**PCC2**	**PCC3**	**PCC4**	**PCC5**
PCP1	Privacy level	16.84	3.45	11	2.21	7.58
	Reputation value	0.4	0.2	0.4	0.4	0.2
PCP2	Privacy level	16	2.81	1.38	9.91	3.9
	Reputation value	0.6	0.2	0.2	0.7	1
PCP3	Privacy level	7.94	5.18	5.18	16.36	6.3
	Reputation value	0.2	0.4	0.4	0.2	0.2

These payments are calculated using formula 1 based on the privacy levels and reputation values given in Table IV. As we can see, the higher payments are given to vehicles with high reputation values and high privacy levels. Thus, to increase their payments, vehicles always try to increase their reputation values and participate in PCPs even if their privacy levels are high. Figure 8 compares the average payment and privacy level received by PCCs and the cost paid by a PCR under both SSCs and OSCs. As shown in Table V, if a PCR performs a PCP under an SSC, only five PCCs will cooperate with it. However, if the same PCR performs a PCP under OSC, three other PCRs and 27 PCCs will participate in this PCP. Figure 8 (a) compares the average payment received by a PCC both under a SSC and an OSC. As we can see, the average payment received by a PCC is higher under an SSC than under an OSC. However, as shown in Figure 8 (b), the privacy level obtained by a PCC under an OSC is higher than the obtained under an SSC.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2021.3083477, IEEE Transactions on Vehicular Technology

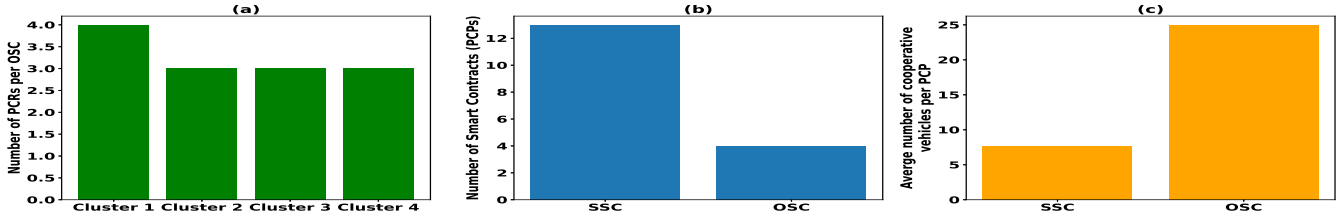

Fig. 7: Comparison between PCPs running under SSCs and PCPs running under OSCs. (a) The distribution of PCRs over clusters; (b) The number of generated smart contracts; (c) The average number of cooperative vehicles per PCP; (ρ =100 veh/km and $size_{CZ}$ = 60 m)

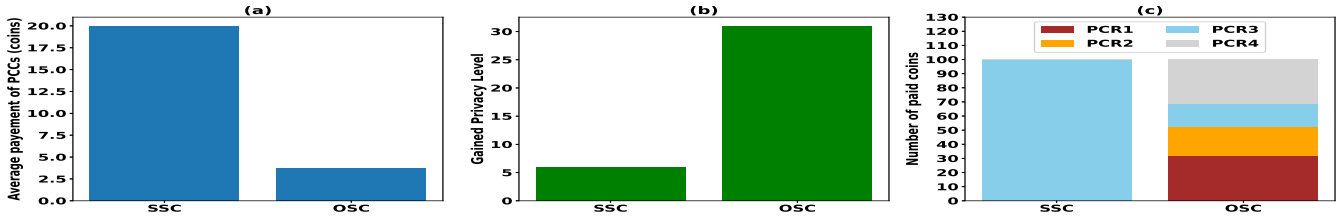

Fig. 8: Comparison between the average payment (a) and privacy level (b) received by PCCs, and the cost paid by a PCR (c) participating in both PCPs under SSCs and PCPs under OSCs; (ρ =100 veh/km and $size_{CZ}$ = 60 m)

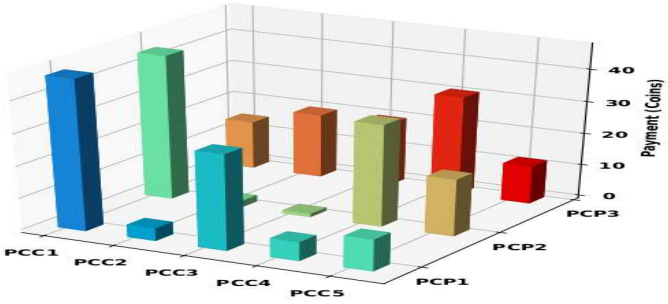

Fig. 9: The payment of five vehicles in three different PCPs (ρ = 100 veh/km, $size_{CZ}$ = 60 m, C = 100 $coins$)

TABLE V: Comparison of the number of PCR(s) and PCCs under a SSC and a OSC

Total price (C)	Type of Smart Contract	PCR(s)	PCCs
100	SSC	1	5
	OSC	4	27

Figure 8 (c) compares the price paid by one PCR (PCR3) under both an SSC and an OSC. As we see, the price paid by PCR3 under an OSC is more 80% lower than the price paid under an OSC.

C. Blockchain analysis

In this section, we first evaluate the consensus time in the consortium blockchain and carry out an analytic evaluation for selecting the first leader in the set of the consensus members. Then, we study the implementation of the proposed scheme in a real case. To calculate the average time to reach the consensus, we run an implementation of the DBFT consensus protocol developed using Python programming language in a machine equipped with a CPU (Intel i5 2.6 GHz) and 8 GO of RAM.

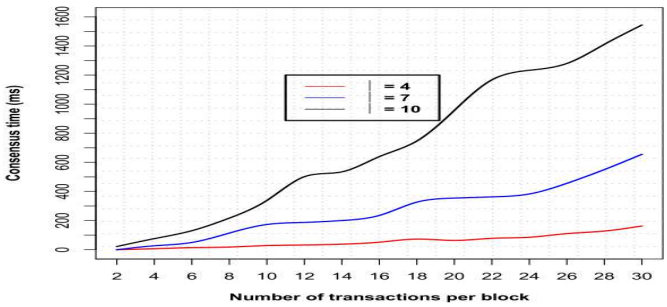

Fig. 10: The average consensus time in the consortium blockchain (milliseconds)

Figure 10 illustrates the consensus time related to one PCP. In this Figure, for each curve, we fixed the number of consortium members and variated the number of transactions up 30. These transactions are generated after the execution of a SC to transfer coins/or to apply penalties. The reason why we limited the number of transactions to 30, is that the number of transactions is depending on the number of cooperative vehicles inside the PCPs. Indeed, as Figures 4 and 5 show the max number of cooperative vehicles in a PCP can achieve 13 when $size_{cz}$ = 90m. Also, since an OSC can involve multiple PCPs and PCRs, this number of transactions can reach 30. Figure 10 shows a linear increase in the consensus time with the number of transactions. They also show that the consensus time increases with the number of consensus members. However, the consensus is reached in a short time. Indeed, it takes only 1.6 seconds to reach a consensus for a

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2021.3083477, IEEE Transactions on Vehicular Technology

block with 30 transactions and 10 consensus members.

In the following, we consider a consortium blockchain consists of four consortium members under different traffic densities and CZ sizes. We have run a numeral evaluation to calculate their utility values for determining the first leader that initiates the consensus process. In this evaluation, the fixed parameters are set as follows: max_{tp} = 1 s, max_{tc} = 0.14 s, nb_{prop} = 1, and C = 100 $coins$. Figure 11 shows the utility values of the consensus members calculated using the formula 5. The obtained results show that BS3 has the highest utility value among the consensus members. Thus, it will be select as the first leader.

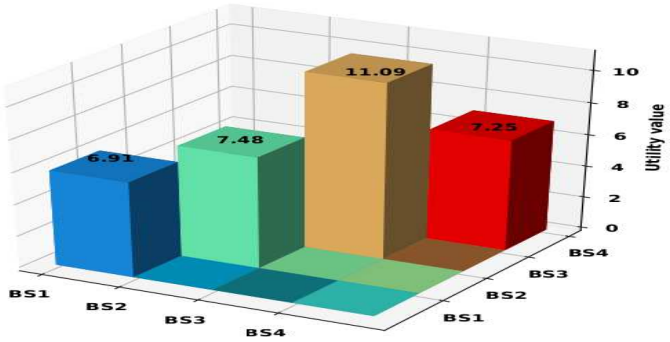

Fig. 11: The utility value of four consensus members under different traffic densities and candidature zone sizes.

We also consider Luxembourg as a case of the application of our scheme. In 2020, Luxembourg has started the deployment of 5G. The first stage of deployment will mainly cover Luxembourg City [41]. The official geoportal of Luxembourg shows the distribution of BSs in Luxembourg city [42]. Among around 750 BSs deployed in the whole country, around 100 BSs are deployed in Luxembourg City. The city also counts around 288 thousand vehicles between local vehicles, buses, and transit vehicles circulating in the city over the 24 hours [43]. During the peak hour (8 am) more than 4.7 thousand vehicles can be found on the road, while at midnight (lull hour), around 700 vehicles left on roads. In the following, we estimate the number of requests that arrives from PCRs, the number of PCPs executed, and the consensus time by BS. We consider that vehicles are uniformly distributed over BSs. Therefore at the peak hour, we count around 470 vehicles per BS, while at the lull hour, only 70 vehicles can be found under a BS. We also consider that the privacy levels of vehicles are distributed according to a normal distribution with a mean equals the common desired privacy level of drivers. Given that vehicles tend to request for a PCP if their privacy levels go below the average, half of the vehicles under a BS can request for PCP (235 PCRs at the peak hour, 70 PCRs at the lull hour). However, as shown in Figure 12, the number of PCRs' requests that can arrive at the BS depends on the probability that PCRs request support from the scheme. In addition, the number of PCPs to be executed is limited to the number of vehicles monitored by the BS. Indeed, as shown in Figure 4, if we consider $size_{CZ}$ = 60 m and the traffic density = 100 veh/km, the number of collaborative of vehicle inside the PCPs equals to 7. Thus at the peak hour, only 68 PCPs need be executed to ensure privacy protection, while in the lull hour 10 PCPs need to be executed. As shown in Figure 12, if the request probability of PCRs is around 0.3, all PCPs are executed with the support of the scheme.

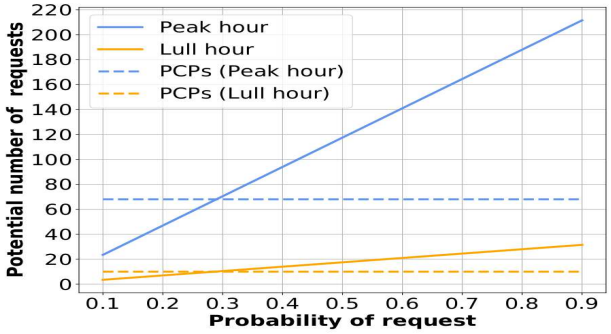

Fig. 12: Potential number of PCRs' requests versus the probability of request

We also estimate consensus time under SSC and OSC. We consider that 10% of 100 the BSs deployed in the city are part of the consortium blockchain, which explains the number of consortium members considred in Figure 10. Table VI, compares cumulative consensus time under SSCs with the consensus time under an OSC for peak hour and the lull hour. The results show that in the peak hour, OSC takes a longer consensus time compared to SSCs. However, in lull hour the results show the consensus time under an OSC is close to SSCs. The consensus time can be enhanced further in the real deployment of the scheme with the high performance of 5G BS [44], and the ultra-low latency offered by 5G networks.

TABLE VI: Comparison of consensus time for peak and lull hour under SSCs and a OSC

Type of Smart Contract	Peak hour	Lull hour
SSCs	14.28s	2.10s
OSC	49.58s	5.19s

D. Security Game Model

In this section, we propose a security game model to capture different attacker behaviors. We consider two kinds of players, the security agent that is activated at each vehicle and BS to monitor its neighbors vehicles and BS, and malicious vehicles and infected BSs that execute the attacks defined in subsection III.C including internal and DoS. We note that, Ψ_j and Ψ_i are the security agent and attacker players, respectively, where i $\in$ {1,...,N}, and N is the number of attackers that attack the player Ψ_j, and j $\in$ {1,..., M}, and M is the number of security agents that monitor the player Ψ_i. The players Ψ_i and Ψ_j have a set of strategies defined respectively as $\zeta_{(\Psi_i)} = \{\Psi_i^{i'} | i' = 1, \ldots, n'\}$ and $\zeta_{(\Psi_j)} = \{\Psi_j^{j'} | j' = 1, \ldots, m'\}$, where n' and m' are the maximum number of strategies. The strategies of player Ψ_i are the number of attacks executed by the attackers against

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2021.3083477, IEEE Transactions on Vehicular Technology

the legitimate vehicles and BSs. The strategies of player Ψ_j are the number of monitored vehicles and BSs that are suspected to execute the malicious behaviors cited above. Let, $x_{i'}$ be the probability of player Ψ_i to execute the strategy $\Psi_i^{i'}$ and $y_{j'}$ be the probability of player Ψ_j to launch the strategy $\Psi_j^{j'}$; where $\sum_{i'=1}^{n'} x_{i'} = 1$ and $\sum_{j'=1}^{m'} y_{j'} = 1$. The utility functions of the players Ψ_i and Ψ_j are shown in formulas 12 and 13.

$$u_{\Psi_j}^t(t) = y_{j'} * \left(\frac{ED^t - (FP^t + FN^t)}{T^t} \right) - Cost_{\Psi_j}. \quad (12)$$

$$u_{\Psi_i}^t(t) = x_{i'} * \left(\frac{(FP^t + FN^t) - ED^t}{T^t} \right) - Cost_{\Psi_i} \quad (13)$$

Here, ED^t is the expected detection rate against the attackers that suspected to occur, and FP^t and FN^t are respectively the false positive and false negative rates against the suspected attacks, e.g., Ψ_j suspects the legitimate non-cooperative vehicle (and BS) as an attacker and vice versa. T^t is the total number of malicious vehicles (and BSs) that occur and target the player Ψ_j. $Cost_{\Psi_j}$ is the required cost of player Ψ_j to achieve a high level of security, high ED^t, while generating low FN^t and FP^t. $Cost_{\Psi_i}$ is the required cost of player Ψ_i to execute attacks strategies ζ_{Ψ_i} against the player Ψ_j. Here, $Cost_{\Psi_j}$ and $Cost_{\Psi_i} \in$]0,1]. In the proposed non-cooperative game, the players Ψ_j run their optimal strategies $\Psi_j^{*j'}$ for detecting the malicious players Ψ_i by taken into account the best responses of these non-cooperative players Ψ_i, while the malicious vehicles (and BSs) Ψ_i run their optimal strategies $\Psi_i^{*i'}$ for executing the attacks by taken into account the best responses of the cooperative players Ψ_j. It is noted, the best response of player Ψ_j is the accuracy of detecting the attacks, i.e., the ED^t is high and the best response of player Ψ_i is executing the attacks against Ψ_j, without being detected, i.e., the FP^t and FN^t are high. Therefore, the strategies couple ($\Psi_j^{*j'}$, $\Psi_i^{*i'}$) executed by the players Ψ_j and Ψ_i are determined by computing the optimal coordinates (δ^{*1}, δ^{*2}) defined as a Nash Equilibrium (NE) point [45], which is equal to:

$$\begin{aligned} \delta^{*1} &= \underset{y_{j'}}{\operatorname{argmax}}\, u_{\Psi_j}^t(t) \\ \delta^{*2} &= \underset{x_{i'}}{\operatorname{argmax}}\, u_{\Psi_i}^t(t) \end{aligned} \quad (14)$$

From formula 14, we conclude that when $u_{\Psi_i}^t(t)$ is equal to $\operatorname{argmax}_{x_{i'}} u_{\Psi_i}^t(t)$, the attacker Ψ_i executes an attack such malicious PCR, PCC or BS against the player Ψ_j. In this case, the security agent Ψ_j categorizes the player Ψ_i as a malicious vehicle (or malicious BS), i.e., $u_{\Psi_j}^t(t)$ is equal to $\operatorname{argmax}_{y_{j'}} u_{\Psi_j}^t(t)$. As shown in Figure 13, we vary the number of iterations from 10 to 40 iterations, where at each iteration each player aims to maximize its utility function and minimize the utility function of its opponent, i.e., the security agent aims to decrease ED^t, while FP^t and FN^t are taken into account and attacker focus to increase the FP^t and FN^t and decrease ED^t. By increasing the number of iterations, we found that there is a point of intersection of two curves (related to the functions $u_{\Psi_i}^t(t)$ and $u_{\Psi_j}^t(t)$, which is defined a Nash equilibrium point, ($u_{\Psi_j}^t(t)$, $u_{\Psi_i}^t(t)$). Therefore, when this equilibrium point is reached the security agent categorizes the malicious vehicles (or BS) with a high accuracy, i.e., detection rate and false positive rates are equals respectively to 100% and 0%.

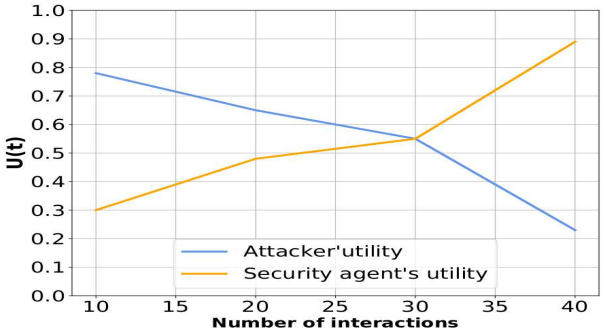

Fig. 13: Nash equilibrium solution

VIII. Discussion

In this section, we discuss the incentive techniques proposed by our scheme. We then perform security, privacy and fairness analyses. Finally, we compare between SSCs and OSCs and give some recommendations.

A. Incentive techniques

In our scheme, several incentive techniques have been proposed to stimulate non-cooperative vehicles. As shown in Algorithm 1 (step 9), the SC pushes vehicles to keep their reputation values positive to be able to request for a PCP. In addition, since the payments received by PCCs depend on their reputation values and their privacy levels, vehicles always try to increase their reputation values and cooperate even when their privacy levels are high to get higher payments. OSCs are another reason for vehicles to increase their reputation values through cooperation. Indeed, since the price paid by PCCs under OSCs depends on their reputation, vehicles should maintain their reputation values high to pay less when OSCs are performed. Moreover, since the consensus members are selected based on their utility, BSs will work to execute efficient PCPs to participate in the consensus processes and get coins. Also, the results show that our scheme allows more cooperative vehicles at PCPs than MPSVLP. Indeed, while our scheme can motivate more than six vehicles when $size_{CZ}$ equals 60m, MPSVLP can only motivate between three and four vehicles in a CZ of more than 100 m. Our scheme fulfills incentive and budget proprieties: (i) Individual Rationality (IR): since both PCR and PCCs will receive positives utilities in terms of privacy protection level and monetary gain respectively, (ii) Incentive compatibility (IC): since the payment of PCCs is calculated with the same formula (formula 1) whatever the smart contract is, and (iii) Budget balance (BB): since the request for a PCP is controlled by the vehicle according to its budget. In other words, vehicles can manage their requests for PCPs to ensure that their generated profits are always positive.

B. Security, Privacy & Fairness Analyses

Our scheme provides a set of security checks to thwart attackers defined in Section III. For thwarting malicious PCRs, the SC verifies the PCR's balance every time it receives its request for a PCP. If a PCR sends a request without having enough coins in its balance, the SC refuses the request and decreases the PCR's reputation value. The smart contract also moves a deposit from the PCR's balance to the contract address for ensuring the payments of PCPs. Moreover, a penalty is applied if the PCR violates any contract clause. For thwarting malicious PCCs, the SC requires PCCs to move deposits from their balances to the contract address. Penalties and reputation decreases are applied to PCCs in the case of non-respect of SC clauses. Our scheme is also thwarting malicious BSs, which try to tamper data to increase their utility for being consensus members. Indeed, our scheme stores all relevant data such as reputation values, scores, and utility values in the blockchain, which cannot be modified without a consensus process. Moreover, our scheme is based on a resilience consensus protocol where the consensus can be reached even that almost the third of BSs are faulty/malicious nodes. Attackers with fake identifiers cannot join the consortium, since members should be authenticated with the CA. Furthermore, our scheme ensures accurate detection of internal and DoS attacks thanks to a game theory-based defense mechanism proposed in subsection VII-D. On the other hand, location privacy preservation of vehicles in the consortium blockchain is ensured since pseudonyms are used as sources of transactions and as account addresses as well. PCRs cannot link between two consecutive pseudonyms of PCCs. However, BSs can only link between two pseudonyms of the same vehicle for matching the feedback messages received by the $invoke$ function with the SC. But since not all PCPs are executed with the support of our scheme, BSs cannot continually link all the pseudonyms of the vehicles. In addition, in our scheme, the accountability is maintained since only the CA can link between real identifiers of vehicles and their corresponding pseudonyms. Our scheme also ensures fairness at different levels: (i) As shown in formula 1, the payment of PCC is performed according to its current privacy level and reputation value, which ensures a fair payment system that rewards PCCs for their sacrifice and their cooperative behavior, (ii) as shown in formula 3, the contribution of each PCR in the total price of the PCP is computed according to its reputation, which is also fair since it makes sure that PCCs with higher reputation values contribute less in the total price, and (iii) As shown in formula 5, the calculation of utility of BS takes into the account the reputation values of vehicles, which ensures fair weights of vehicles' feedback used in the calculation of utility values. On the other hand, there is no fairness issue if certain vehicles travel more than the others. Since as long as vehicles are traveling, they will have more opportunities to participate PCPs but also their privacy levels will decrease.

C. SSC vs OSC

Our scheme proposes two types of smarts contracts: SSCs and OSCs. Our evaluation results show that OSCs allows reducing the number of SCs managed by the scheme and decreasing the costs paid by PCRs compared to SSCs. They also show that while the payment received by a PCC under an OSC is lower than the payment received under an SSC, the location privacy level is better under an OSC. However, the creation of OSCs takes more longer than SSCs since BSs need to wait a certain time collecting requests for PCRs, which may result in an excessive delay for executing PCPs on time. Therefore, our recommendations are to adapt the duration of PCRs' requests collection (ΔT_1) according to the number of requests and the maximum time allowed to execute the PCP. The results also show that OCSs have a longer consensus time than SSCs, especially peak traffic hours.

IX. Conclusion

This paper proposed a novel a consortium blockchain-based cooperative location privacy scheme for 5G-enabled Vehicular Fog computing. Leveraging SCs and combining monetary and reputation incentive techniques, our scheme ensures successful, trusted, and secure Pseudonym Changing Processes (PCPs). Our scheme is privacy-persevering and leverages a resilient and lightweight consensus protocol, which provides fast and reliable consensus processes. Moreover, our scheme provides standard (SSCs) and optimized (OSCs) SCs. While OSCs can provide better location privacy levels and efficient monetary cost management, the delay of the creation of OSCs can lead to PCPs' failures. In this vein, our future work will investigate adaptive techniques that consider additional parameters such as mobility and traffic density in the creation of OSCs.

Acknowledgment

This work was supported by the 5G-DRIVE project and 5G-MOBIX project. Both projects have received funding from the European Union's Horizon 2020 research and innovation programme under grant agreements No 814956 and No 825496 respectively. Content reflects only the authors' view and European Commission is not responsible for any use that may be made of the information it contains

References

[1] M. Boban, A. Kousaridas, K. Manolakis, J. Eichinger, and W. Xu, "Use cases, requirements, and design considerations for 5G V2X," *arXiv preprint arXiv:1712.01754*, 2017.

[2] L. Liu, C. Chen, Q. Pei, S. Maharjan, and Y. Zhang, "Vehicular edge computing and networking: A survey," *arXiv preprint arXiv*, 2019.

[3] Y. Dai, D. Xu, S. Maharjan, and Y. Zhang, "Joint load balancing and offloading in vehicular edge computing and networks," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4377–4387, 2018.

[4] Y. Dai, D. Xu, S. Maharjan, G. Qiao, and Y. Zhang, "Artificial intelligence empowered edge computing and caching for internet of vehicles," *IEEE Wireless Communications*, vol. 26, no. 3, pp. 12–18, 2019.

[5] A. Boualouache, R. Soua, and T. Engel, "Toward an SDN-based Data Collection Scheme for Vehicular Fog Computing," in *IEEE International Conference on Communications ICC'2020*, 2020.

[6] J. Kang, R. Yu, X. Huang, M. Wu, S. Maharjan, S. Xie, and Y. Zhang, "Blockchain for secure and efficient data sharing in vehicular edge computing and networks," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4660–4670, 2018.

[7] D. Eckhoff and C. Sommer, "Driving for Big Data? Privacy Concerns in Vehicular Networking," *IEEE Security and Privacy*, vol. 12, no. 1, pp. 77–79, February 2014.

[8] F. Dressler, F. Kargl, J. Ott, O. K. Tonguz, and L. Wischhof, "Research challenges in intervehicular communication: lessons of the 2010 dagstuhl seminar," *IEEE Communications Magazine*, vol. 49, no. 5, pp. 158–164, 2011.
[9] IEEE, "IEEE Standard for Wireless Access in Vehicular Environments Security Services for Applications and Management Messages," *IEEE Std 1609.2-2013 (Revision of IEEE Std 1609.2-2006)*, pp. 1–289, 2013.
[10] ETSI, "Intelligent transport systems (its); security; trust and privacy management," *Standard, TC ITS*, 2018, technical specification, ETSI.
[11] A. Boualouache and S. Moussaoui, "TAPCS: Traffic-aware pseudonym changing strategy for VANETs," *Peer-to-Peer networking and Applications*, vol. 10, no. 4, pp. 1008–1020, 2017.
[12] A. Boualouache, S.-M. Senouci, and S. Moussaoui, "A survey on pseudonym changing strategies for vehicular ad-hoc networks," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 1, pp. 770–790, 2017.
[13] J. Freudiger, M. H. Manshaei, J.-P. Hubaux, and D. C. Parkes, "Non-cooperative location privacy," *IEEE Transactions on Dependable and Secure Computing*, vol. 10, no. 2, pp. 84–98, 2012.
[14] S. Du, H. Zhu, X. Li, K. Ota, and M. Dong, "Mixzone in motion: achieving dynamically cooperative location privacy protection in delay-tolerant networks," *IEEE transactions on vehicular technology*, vol. 62, no. 9, pp. 4565–4575, 2013.
[15] R. Lu, X. Lin, T. H. Luan, X. Liang, and X. Shen, "Pseudonym changing at social spots: An effective strategy for location privacy in VANETs," *IEEE transactions on vehicular technology*, vol. 61, no. 1, pp. 86–96, 2011.
[16] B. Ying, D. Makrakis, and Z. Hou, "Motivation for protecting selfish vehicles' location privacy in vehicular networks," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 12, pp. 5631–5641, 2015.
[17] A. Boualouache, S.-M. Senouci, and S. Moussaoui, "PRIVANET: An efficient pseudonym changing and management framework for vehicular ad-hoc networks," *IEEE Transactions on Intelligent Transportation Systems*, 2019.
[18] J. R. Marden and A. Wierman, "Overcoming limitations of game-theoretic distributed control," in *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*. IEEE, 2009, pp. 6466–6471.
[19] O. Novo, "Blockchain meets IoT: An architecture for scalable access management in IoT," *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 1184–1195, 2018.
[20] A. Dorri, M. Steger, S. S. Kanhere, and R. Jurdak, "Blockchain: A distributed solution to automotive security and privacy," *IEEE Communications Magazine*, vol. 55, no. 12, pp. 119–125, 2017.
[21] S. Bao, Y. Cao, A. Lei, P. Asuquo, H. Cruickshank, Z. Sun, and M. Huth, "Pseudonym Management Through Blockchain: Cost-Efficient Privacy Preservation on Intelligent Transportation Systems," *IEEE Access*, vol. 7, pp. 80 390–80 403, 2019.
[22] A. Lei, Y. Cao, S. Bao, D. Li, P. Asuquo, H. Cruickshank, and Z. Sun, "A blockchain based certificate revocation scheme for vehicular communication systems," *Future Generation Computer Systems*, 2019.
[23] N. U. Hassan, C. Yuen, and D. Niyato, "Blockchain technologies for smart energy systems: Fundamentals, challenges, and solutions," *IEEE Industrial Electronics Magazine*, vol. 13, no. 4, pp. 106–118, 2019.
[24] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," Manubot, Tech. Rep., 2019.
[25] J. Freudiger, M. Raya, M. Felegyhazi, P. Papadimitratos, and J. Hubaux, "Mix-zones for location privacy in vehicular networks," in *Proceedings of the First International Workshop on Wireless Networking for Intelligent Transportation Systems (Win-ITS)*, 2007.
[26] A. Boualouache and S. Moussaoui, "Urban pseudonym changing strategy for location privacy in vanets," *International Journal of Ad Hoc and Ubiquitous Computing*, vol. 24, no. 1-2, pp. 49–64, 2017.
[27] M. Gerlach and F. Guttler, "Privacy in vanets using changing pseudonyms-ideal and real," in *2007 IEEE 65th Vehicular Technology Conference-VTC2007-Spring*. IEEE, 2007, pp. 2521–2525.
[28] A. Wasef and X. Shen, "Rep: Location privacy for vanets using random encryption periods," *Mobile Networks and Applications*, vol. 15, no. 1, pp. 172–185, 2010.
[29] Y. Dai, D. Xu, K. Zhang, S. Maharjan, and Y. Zhang, "Deep reinforcement learning and permissioned blockchain for content caching in vehicular edge computing and networks," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 4, pp. 4312–4324, 2020.
[30] A. Rashid and M. J. Siddique, "Smart Contracts Integration between Blockchain and Internet of Things: Opportunities and Challenges," in *2019 2nd International Conference on Advancements in Computational Sciences (ICACS)*. IEEE, 2019, pp. 1–9.
[31] M. B. Mollah, J. Zhao, D. Niyato, Y. L. Guan, C. Yuen, S. Sun, K.-Y. Lam, and L. H. Koh, "Blockchain for the internet of vehicles towards intelligent transportation systems: A survey," *IEEE Internet of Things Journal*, 2020.
[32] L. Li, J. Liu, L. Cheng, S. Qiu, W. Wang, X. Zhang, and Z. Zhang, "Creditcoin: A privacy-preserving blockchain-based incentive announcement network for communications of smart vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 7, pp. 2204–2220, 2018.
[33] X. Zhang and X. Chen, "Data security sharing and storage based on a consortium blockchain in a vehicular ad-hoc network," *IEEE Access*, vol. 7, pp. 58 241–58 254, 2019.
[34] Z. Su, Y. Wang, Q. Xu, M. Fei, Y.-C. Tian, and N. Zhang, "A secure charging scheme for electric vehicles with smart communities in energy blockchain," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4601–4613, 2018.
[35] S. Wang, X. Huang, R. Yu, Y. Zhang, and E. Hossain, "Permissioned blockchain for efficient and secure resource sharing in vehicular edge computing," *arXiv preprint arXiv:1906.06319*, 2019.
[36] A. Rodriguez and A. Laio, "Clustering by fast search and find of density peaks," *Science*, vol. 344, no. 6191, pp. 1492–1496, 2014.
[37] M. Castro, B. Liskov *et al.*, "Practical Byzantine fault tolerance," in *OSDI*, vol. 99, no. 1999, 1999, pp. 173–186.
[38] C. Sommer, R. German, and F. Dressler, "Bidirectionally coupled network and road traffic simulation for improved ivc analysis," *IEEE Transactions on Mobile Computing*, vol. 10, no. 1, pp. 3–15, Jan 2011.
[39] D. Krajzewicz, J. Erdmann, M. Behrisch, and L. Bieker, "Recent development and applications of sumo – simulation of urban mobility," *International Journal on Advances in Systems and Measurements*, vol. 5, no. 3, pp. 128–138, 2012.
[40] S. Krauß, P. Wagner, and C. Gawron, "Metastable states in a microscopic model of traffic flow," *Physical Review E*, vol. 55, no. 5, p. 5597, 1997.
[41] Incites, T. the department of Media, and D. Policy, "Luxembourg 5g strategy expert report," 2018.
[42] *https://map.geoportail.lu/theme/cadastre_hertzien*, accessed November 11, 2020.
[43] L. Codecá, R. Frank, S. Faye, and T. Engel, "Luxembourg sumo traffic (lust) scenario: Traffic demand evaluation," *IEEE Intelligent Transportation Systems Magazine*, vol. 9, no. 2, pp. 52–63, 2017.
[44] *https://www.intel.com/content/www/us/en/products/processors/atom/p-series/atom-p-5* accessed November 12, 2020.
[45] L. Zhang and E. Hemberg, "Investigating algorithms for finding nash equilibria in cyber security problems," in *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, 2019, pp. 1659–1667.

Abdelwahab Boualouache (M'15) is a research associate at the University of Luxembourg. He received the Ph.D. degree in computer science from USTHB University, Algiers, Algeria in 2016. His current research interests include security and privacy in connected vehicles and privacy-preserving collaborative learning solutions for 5G.

Hichem Sedjelmaci is a Senior Research Engineer and Projects Manager in Cyber Security and AI, at Orange Labs. He received the PHD degree in telecommunication systems from Tlemcen university, Algeria in 2013 and from university of Burgundy, France in 2019. His current research interests include Cyber security, ML, and ITS

Thomas Engel is Professor for Computer Networks at the University of Luxembourg. He received the title Dr. rer. nat form the University of Saarbruecken, Germany in 1996. His SECAN-Lab team deals with performance, privacy and identity handling in Next Generation Networks.