A Light Blockchain-Powered Privacy-Preserving Organization Scheme for Ride Sharing Services

Mohamed Baza*, Mohamed Mahmoud*, Gautam Srivastava†, Waleed Alasmary‡, Mohamed Younis§
*Department of Electrical and Computer Engineering, Tennessee Tech University, Cookeville, TN, USA
†Department of Mathematics and Computer Science, Brandon University, Manitoba, Canada
‡Department of Computer Engineering, Umm Al-Qura University, Makkah, Saudi Arabia
§Department of Computer Science and Electrical Eng., University of Maryland, Baltimore County, MD 21250

***Abstract*—Ride-sharing is a service that enables drivers to share their trips with other riders, contributing to improving traffic congestion as well as assist in reducing Carbon Dioxide (CO2) emission and fuel consumption. It has come to the forefront in recent years as a Green service in large cities. However, the majority of existing ride-sharing services rely on a central third party, which makes them subject to a single point of failure and privacy disclosure concerns by both internal and external attackers. Moreover, they are vulnerable to distributed denial of service (DDoS) and Sybil attacks due to malicious users. There is also high service fees paid to the ride-sharing service provider. In this paper, we propose to decentralize ride-sharing services based on a public Blockchain. Our scheme enables drivers to propose ride-sharing services without relying on a trusted third party. To preserve location privacy, riders send cloaked ride requests to hide their exact pick-up/drop-off locations, and departure/arrival dates. Then, by using an off-line matching technique, drivers sends their offers encrypted to ensure data confidentiality. Upon receiving the ride-offers, the rider can find a ride match using some heuristics as well as the bid price included in the offer. To preserve anonymity, riders/drivers use pseudonyms that change per trip to ensure unlinkabilty. We envision the application of this technology in Green Internet of Things connected smart cities, where ride sharing services are common. Finally, we implement our scheme and deploy it in a test net of Ethereum. The experimental results show the applicability of our protocol.**

***Index Terms*—Ride sharing services, Blockchain, Smart contract, Ethereum, green, smart cities**

I. Introduction

Over the last few years, ride-sharing services (RSS) have emerged as an alternative transportation service that allows people to use personal cars efficiently. This coincides with a recent push to incorporate green technology into our ever-expanding vehicle use in large cities. When providing a ride-sharing service (RSS), a driver shares his vacant car seats with other riders who are traveling in the same direction contributing to several benefits to the individual and the community at large. Some of these benefits include increasing occupancy rates in vehicles, sharing travel costs, extending social circles, lowering road traffic, and reducing both fuel consumption and air pollution [1]. RSS can easily be considered a green technology and a huge step forward for smart cities. Across the world, many providers for online ride-sharing services such as Flinc, UberPool, Lyft Line and Blablacar have emerged. According to [2], the ride sharing market is projected to reach USD 218 billion by 2025.

A ride-sharing service strives to match drivers with appropriate riders according to their respective ride offers (i.e., planned trips) and ride requests (i.e., desired trips). To enable a ride-sharing service, users (i.e., drivers and riders) have to share with a service provider the trip detail information, including departure time and location, and the destination. The service provider works as a middleman to facilitate the communication between the system users and usually charges a commission for each successful ride-share. However, since user's data is stored at the central service provider, the system becomes vulnerable to a single point of failure. Particularly, if the service provider is compromised, the service can be interrupted and the data can be disclosed, altered, or even deleted. For instance, Uber experienced a tremendous data leakage of 57 million customers and drivers for more than a year, and it had to pay 148 million just to settle an investigation to its data breach [3]. Similarly, in April 2015, due to hardware failure in Uber China, a service outage occurred and passengers were not able to stop their orders at the end of services [4].

In contrast to the traditional client-server model, Blockchain is a verifiable, immutable and distributed ledger that allows mistrusting entities to transact with each other without per-established trust and without relying on a central third party. Smart contracts can be described as autonomous computer programs running on a blockchain network. These programs act as a contract where the terms of the contract can be pre-programmed with the ability to self-execute and self-enforce itself without the need for trusted authorities [5], [6].

In this paper, we propose to implement a ride-sharing system using blockchain and smart contracts to mitigate the single point of failure issues presented in the client-server architecture while preserving riders/drivers privacy. To preserve location privacy, we use the *generalization/cloaking*, so a rider posts a generalized pick-up and drop-off location as well as departure time instead of exact locations and times. Then, interested drivers use an offline matching technique to check if the request falls within their generalized route. They then send the exact information trip data encrypted with rider's public key as well as the shared ride offer price (bid price). Next, a rider can select the best match i.e., the driver to have a trip with based on some heuristics. In essence our system acts as a distributed auction that is handled through the blockchain to ensure transparency. Upon arriving to the pick-up location, and in order to prevent impersonation attacks, where a malicious rider tries to take a ride that was reserved by another rider, both the driver and rider should authenticate each other. This is done using zero-knowledge where the rider should prove the possession of

978-1-7281-5207-3/20/$31.00 ©2020 IEEE

the private key corresponding to the public key used in the reservation. Finally, we have implemented and deployed our scheme in an Ethereum test net. The results indicate that the practicality of our scheme.

The rest of the paper is organized as follows. The assumptions and threat models are discussed in Section II. Our system is detailed in Section III. We present the security, privacy, and computation complexity analysis of our scheme in Section IV. Section V discusses the previous research work. Finally, we give concluding remarks in Section VI, followed by an acknowledgement in Section VII.

II. System and Threat models

As illustrated in Fig. 1, the considered system model has two main entities.

- *Blockchain.* At the heart of our system is the blockchain network that handles all ride sharing transactions. We opt for a permissionless blockchain where everyone can use the system to either act as a driver or rider. The ride sharing business logic is defined in a smart-contract and executed by the blockchain network. The blockchain is also used for P2P payments to allow the exchange of currency between the different system users. Thus, we select Ethereum, the most popular open blockchain platform for smart-contracts, to implement and evaluate our proposed decentralized ride sharing service.
- *Driver and rider.* The rider is an entity that post requests the ride-sharing service to the system. The driver is an entity that wants to share the trip by posting an offer to answer rider's requests. Note that drivers and riders are not required to store a complete copy of the blockchain locally. Instead, they can run on top of so-called light-weight nodes, which eventually enables them to interact with the network to send transactions or read from the blockchain [7].

For the threat model, we follow the standard blockchain threat model presented in [8], where the blockchain is trusted for execution correctness and availability, but not for privacy. Once deployed, a smart-contract code is visible and readable by anyone and is guaranteed to work as specified, free from tampering. Likewise, any data submitted and stored to the contract can be directly read by all parties of the system as well as any external curious users. We consider global eavesdroppers who can read all previous recorded transactions on the blockchain for riders in order to learn their moving patterns, guess their locations at a specific time or even track them over the time.

III. Proposed Ride-Sharing System

A. Generating Cloaked Planned Trips

In this section, we discuss how the drivers/riders generate their planned trip's data while preserving their privacy. A generalization technique also known as spatial cloaking, is used for this purpose. The main idea of spatial cloaking is to blur the exact users' positions into cloaked regions for location obfuscation. Let us denote $\mathcal{A}$ as the ride sharing region (e.g., a state)

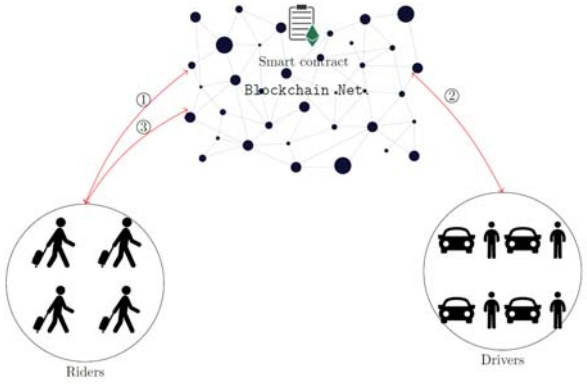

Figure 1: The System architecture of our proposed ride-sharing system: (1) A rider publishes a request contract to the blockchain (2) Drivers send their encrypted offers. (3) A rider selects the best match among the received offers.

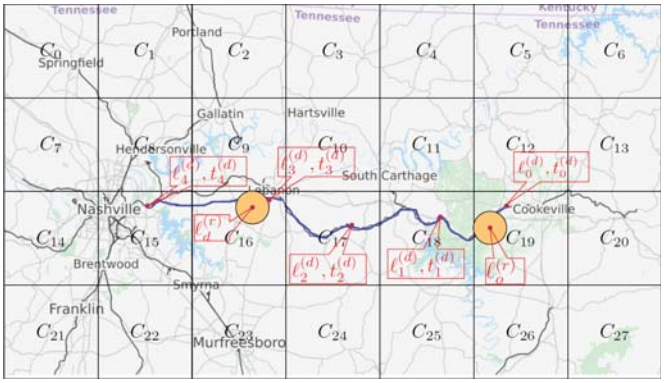

Figure 2: Illustration of dividing the ride sharing area of interest into cells for the state of TN, USA. A driver d's route, in blue, with 5 points, and pickup and drop-off of a rider r.

that is sub-divided into a set of n *cells* $C = \{C_1, C_2, ,,, C_n\}$. Cells could be defined based on geographic area constraints such as districts or neighborhoods in a city, uniform partitions in a map, or using other criteria. Therefore, instead of using the exact pick-up/drop-off location, riders can only submit their respective cell coordinates containing the actual pick-up and drop-off locations. Fig. 2 provides an illustrative example using part of the state of Tennessee, USA, where it is divided into into 27 cells. Similarly, the exact pick-up/drop-off times can be also generalized (hidden) by using temporal cloaking where the actual time is aggregated or summarized by setting *time interval* T. For instance, a driver d who is interested in sharing his vacant seats with others should consider the following steps before sending his offer.

1) Defines the planned trip $\Gamma^{(d)}$

$$\Gamma^{(d)} = \left\{ \left(\ell_0^{(d)}, t_0^{(d)}\right), \ldots, \left(\ell_k^{(d)}, t_k^{(d)}\right), \ldots, \left(\ell_n^{(d)}, t_n^{(d)}\right) \right\}$$

that consists of the exact departure location $\ell_0^{(d)}$, departure time $t_0^{(d)}$, destination $\ell_n^{(d)}$ and estimated arrival time $t_n^{(d)}$,

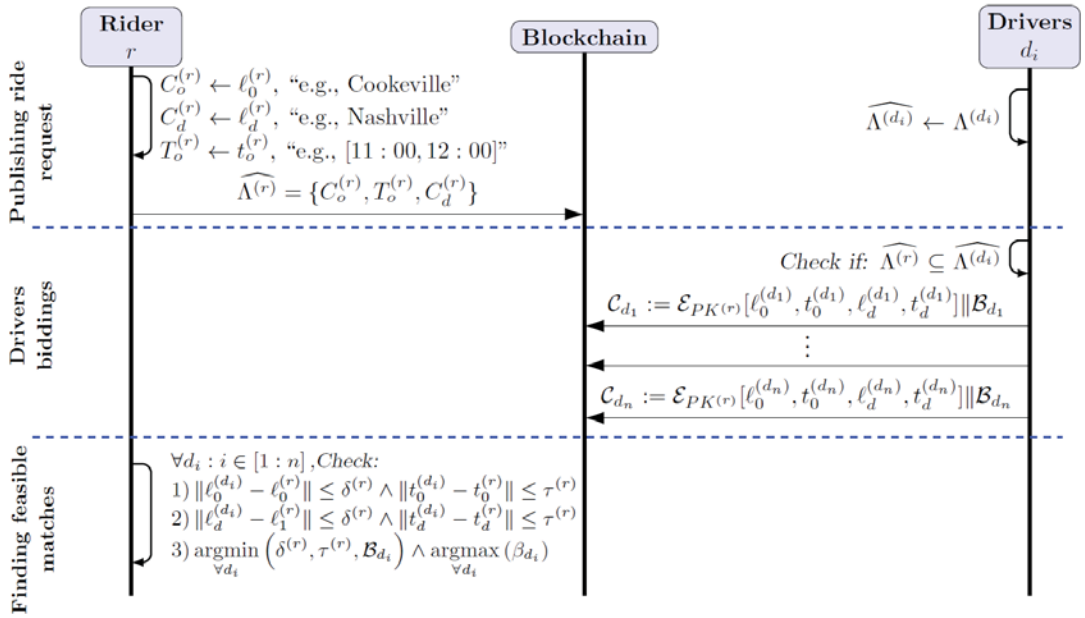

Figure 3: Illustration of the bidding and selection phase.

as well as a sequence of optionally intermediate locations and their corresponding arrival times $\left(\ell_k^{(d)}, t_k^{(d)}\right)$.

2) Then, he maps his trip pick-up and drop-off locations to a cloaked zones (cells) also the exact times to time intervals as follows.

$$\Lambda^{(d)} = \left\{\left(C_0^{(d)}, T_0^{(d)}\right), \ldots, \left(C_n^{(d)}, T_n^{(d)}\right)\right\} \tag{1}$$

where $\left(C_0^{(d)}, T_0^{(d)}\right)$ represents the generalized location and time that corresponding to $\left(\ell_0^{(d)}, t_0^{(d)}\right)$.

3) Finally, to obtain all possible shared trips in his route, he/she create a table of the all trips containing the pick-up cell, drop-off cell and departure time. This can be mathematically represented as follows.

$$\widehat{\Lambda^{(d)}} = \{\left(C_j^{(d)}, T_j^{(d)}, C_k^{(d)}\right)\} \quad 1 \leqslant j \leqslant n \quad j+1 \leqslant k \leqslant n \tag{2}$$

where the number of all possible trips depends on the number of chosen points n and it can be mathematically expressed as [9]

$$\begin{pmatrix} n \\ 2 \end{pmatrix} = \frac{n!}{2!(n-2)!}$$

For a rider r, we denote his ride request as $\Gamma^{(r)} = \left\{\left(\ell_0^{(r)}, t_0^{(r)}\right), \ell_d^{(r)}\right\}$, where $\left(\ell_0^{(r)}, t_0^{(r)}\right)$ represents the departure location and the desired set off time, and $\ell_d^{(r)}$ denotes the drop-off location. Similar to the driver, the rider also convert the request into the generalized form by mapping the pick-up and drop-off locations into the corresponding cells as well as departure time into cloaked time.

$$\widehat{\Lambda^{(r)}} = \{\left(C_o^{(r)}, T_o^{(r)}, C_d^{(r)}\right)\} \tag{3}$$

Note that all the previous steps, including the generation of the drivers' trip (see Table 3) and the rider's planned trip, are done off the blockchain using, for instance, the driver/rider's smart phone devices.

B. Bidding and Selection Phase

In this section, we describe the process of matching rider's requests with driver's offers atop a public blockchain. The different steps required for bidding and selection are illustrated by a schematic diagram in Fig. 3, and detailed in the following sections.

1) Publishing the ride request: First, as a fundamental concept to avoid de-anonymization in the blockchain, every rider r uses for each ride request a new blockchain address $\mathsf{ADD}^{(r)}$ that corresponds to a fresh public-private key pair $(PK^{(r)}, SK^{(r)})$. Then, a rider publishes a ride request that contains his/her generalized pick-up/drop-off $(C_o^{(r)}, C_d^{(r)})$ and generalized time $T_o^{(r)}$. Also, the request should include a deadline of receiving driver's offers. Optionally, the request can include a maximum number of offers to be received. Note that this request should be signed by the temporary private key of the rider and sent to the smart-contract. Once the miners validate the corresponding signature of the rider, the request will be public to all drivers.

2) Submitting drivers' biddings: For a driver (either a company or an individual car owner) that wish to use the ride-sharing service, needs to first get authorization by receiving a public key certificate from the corresponding registration authority (RA), such as the government. A driver d having a unique identity (e.g., license plate number), creates a public-secret key pair $(PK^{(d)}, SK^{(d)})$ and registers at the RA to obtain a certificate binding $PK^{(d)}$ to d.

TABLE 1
DRIVER'S LOCATION COORDINATES & TIMES

Location-time pair	location cooridinates (ℓ)	Time
($\ell_0^{d)}$ $t_0^{d)}$)	36.152627, -85.526534	[8:55]
($\ell_1^{d)}$ $t_1^{d)}$)	36.153735, -85.880843	[9:00]
($\ell_2^{d)}$ $t_2^{d)}$)	36.172584, -86.074477	[9:08]
($\ell_3^{d)}$ $t_3^{d)}$)	36.174801, -86.460372	[9:17]
($\ell_4^{d)}$ $t_4^{d)}$)	36.153830, -86.460372	[9:30]

TABLE 2
GENERALIZED DRIVER ROUTE

Cell (C)	Time Interval (T)
C_{19}	[8:50, 9:00]
C_{18}	[9:05, 9:15]
C_{17}	[9:10, 9:20]
C_{16}	[9:20, 9:25]
C_{15}	[9:20, 9:25]

TABLE 3
DRIVER TRIPS TABLE

C_o	C_d	T_o
$C_{19}^{d)}$	$C_{18}^{d)}$	[8:55, 9:00]
$C_{19}^{d)}$	$C_{17}^{d)}$	[8:50, 9:00]
$C_{19}^{d)}$	$C_{16}^{d)}$	[9:05, 9:15]
$C_{19}^{d)}$	$C_{15}^{d)}$	[9:05, 9:15]
...	...	...
$C_{16}^{d)}$	$C_{15}^{d)}$	[9:25, 9:35]

Drivers can either periodically query the blockchain to ask for new rider's requests or use some out-of-band signaling protocols to get notified each time a new request is published [10]. In order to make an offer, a driver d first verifies if the spatio-temporal attributes of a received request made by a rider r falls within one of his own planned trips, (i.e., $\widehat{\Lambda^{(r)}} \in \widehat{\Lambda^{(d_i)}}$). If one of the requests matches the driver's trip, the driver creates an offer that should include all necessary information for the rider such as, the exact pick-up location and time $\left(\ell_0^{(d_i)}, t_0^{(d_i)}\right)$, the exact drop-off location and time $\left(\ell_d^{(d_i)}, t_d^{(d_i)}\right)$ as well as the offer bid price $\mathcal{B}_{d_i}$ (e.g., charge per mileage). Then, the driver uses the rider' temporary public key (sent with the transaction) to encrypt all above information to obtain $\mathcal{C}_{d_i}$

$$\mathcal{C}_{d_i} = \mathcal{E}_{PK^{(r)}}\left(\ell_0^{(d_i)}, t_0^{(d_i)}, \ell_d^{(d_i)}, t_d^{(d_i)}\right)$$

Where $\mathcal{E}$ is an asymmetric encryption algorithm e.g., RSA, DSA. This is necessary to preserve both the driver and rider privacy. The tuple $\mathcal{C}_{d_i} \| \mathcal{B}_{d_i}$ will be sent by the driver to the smart-contract. Note that the bidding price is not encrypted and made public in order to ensure price competition which guarantee lower prices for the riders.

3) Finding Feasible Matches: After receiving, for instance, n offers $\{\mathcal{C}_{d_1}, \cdots, \mathcal{C}_{d_n}\}$ for the same request, a rider r, retrieves the encrypted offers from the smart-contract and decrypts each of them off the chain using his private key. In order to select an offer that suit the rider, the decrypted offers are evaluated as follows.

1) The driver's pick-up and drop-off match (i) *spatially* by checking whether:

$$|\ell_o^{(d_i)} - \ell_o^{(r)}| \leq \delta^{(r)} \wedge |\ell_d^{(d_i)} - \ell_d^{(r)}| \leq \delta^{(r)}, \qquad (4)$$

where $\delta^{(r)}$ is the maximum distance that the rider can walk to meet the driver's pick-up location, or to reach his final destination.
(ii) *Temporarily* by checking if:

$$|t_o^{(d_i)} - t_o^{(r)}| \leq \tau^{(r)} \wedge |t_d^{(d_i)} - t_d^{(r)}| \leq \tau^{(r)}, \qquad (5)$$

where $\tau^{(r)}$ is the maximum delay that the rider can tolerate before meeting the driver at the pick-up location, or to reach his final destination after being dropped-off.

2) Using $\delta^{(r)}$, $\tau^{(r)}$, and $\mathcal{B}_{d_i}$ the rider is able to select the best offer that match his preferences as follows:

$$\underset{\forall d_i}{\operatorname{argmin}}(\delta^{(r)}, \tau^{(r)}, \mathcal{B}_{d_i})$$

Note that preferences may vary from one rider to another, for instance, some may prefer an offer with a low bid price even if it has a high space slack $\delta^{(r)}$ or waiting time $\delta^{(r)}$. Different from existing centralized approaches, finding feasible ride matches is handled over the blockchain and is therefore, fully transparent.

4) Illustrative example for the selection/matching phase: To better illustrate the selection process, let us assume a driver d_i has a planned trip. The driver's route starts from Knoxville and ends in Nashville with four possible stops points on his route. Note that defining these points depends only on the driver preferences. Table 1 illustrates the coordinates of the stop points with their corresponding expected arrival times which are defined by the driver. As discussed in Sec. III-A, the driver's route should be generalized/mapped as given in Table 2 with respect to a grid overlay (cell based division) of Tennessee given in Fig. 2. Thus, the driver creates trip table that contains possible sub-trips that that may be shared with other riders, as indicated in Table 3.

On the other hand, a rider r who is looking to take a ride with the following attributes:

$$\Gamma^{(d)} = (35{,}222; -100{,}1511, 4548754, 35{,}221; -100{,}1511)$$

publishes a request to the blockchain by sending a generalization version as:

$$\widehat{\Lambda^{(r)}} = \{(Utica, 45222211, Syracuse)\}$$

Each driver, therefore, reads the existing requests on the blockchain to determine to which one they could submit an offer. For instance, based on the trip table of the driver d_i in Table 3, a driver can determine that the rider request is in on route, i.e., the 4th element in Table 3. Thus, he follow up on this request by sending an offer that contains the pick-up, drop-off and time encrypted to the rider. Finally, the rider evaluate the offer and compare it with any other offers to select the best matched one.

C. Driver/rider authentication at the pick-up location

In order to prevent impersonation attacks, where a malicious user tries to take a ride reserved by another requester, driver and

rider must authenticate each other. Specifically, the rider should prove to the driver with zero knowledge that he/she indeed knows the value of private key corresponding to the public key ($PK^{(r)}$) that made the reservation. The driver (the verifier) selects a uniformly random integer $\mathcal{S} \in \mathbb{Z}_p$ as a *challenge* and sends it to the rider (the prover). The rider uses his private key to generate a signature on the challenge ($\sigma_{SK^{(r)}}(\mathcal{S})$) and sends ($\mathcal{S} \| \sigma_{SK^{(r)}}(\mathcal{S})$) to the driver. Finally, the driver verifies if $VerifySig(PK^{(r)}, \sigma_{SK^{(r)}}(\mathcal{S}), \mathcal{S}) = 1$. If the verification succeeds, the driver can start the trip with the rider.

IV. Performance Analysis

In this section we evaluate the performance of our proposed system using real test-net of a public blockchain (Ethereum), followed by security and privacy features.

A. Computation Cost

Ethereum introduced the concept of *gas* to quantify the cost associated with each transaction. The cost is payable using the native Ethereum currency, named *Ether*. Each operation in a smart contract has a fixed cost. For instance, the addition of two variables requires 3 gas, multiplication costs 5 has and computing a SHA3 hash needs 30 gas plus 6 gas for every 256 bits of input [11]. Therefore, to evaluate the cost of the on-chain operations, we are interested in the following metrics.

- *Transaction cost.* is based on the overall gas cost of sending data to the blockchain. Typically, there are four items which make up the full transaction cost; (i) the base constant cost of a transaction, (ii) the cost of a contract deployment, (iii) the cost of every zero byte of data or code in a transaction, (iv) and the cost of every non-zero byte of data or code in a transaction [11].
- *Execution cost.* indicates the portion of gas that is actually spent on running the code included in a transaction by the Ethereum Virtual Machine (EVM) [12].

We have implemented our proposed organization scheme in a smart contract in Solidity 0.4.0, which allows users to design such contracts with private and public methods and has a set of basic data types. The smart contract was deployed into the Kovan blockchain [13].

Fig. 4 reports the gas consumption per trip costs on the rider side. Around 400K gas is required for publishing the bidding contract. 89K gas is required so that a driver can send the encrypted offers as well as the bidding price to the bidding contract. Finally, 200K gas is required to retrieve the drivers' offers. These costs can be translated to compute direct monetary cost of our scheme. Table 4 gives the estimated total cost of driver versus the number of riders, given gas price of 0.5 Gwei and Ether price $208 as of Sept. 20th, 2019 [14]. Having 1000 trips, the driver costs about $ 15. The results clearly show that the cost is low and very affordable for the end users comparing to current ride-sharing service providers.

B. Security/Privacy Analysis

Our proposed scheme presents the following unique features:

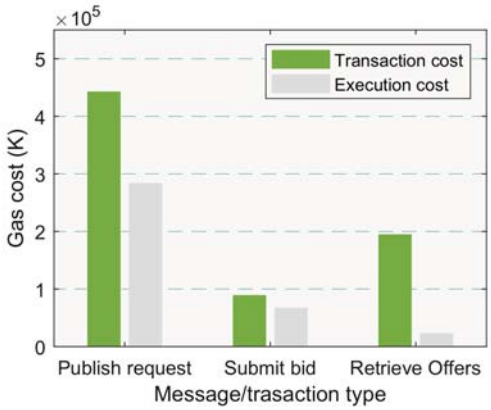

Figure 4: Estimated on-chain computation cost.

Table 4: The momentary cost of drivers versus number of trips

Number of trips	1	10	100	1000
Cost (USD)	0.02	0.16	1.6	15.6

Decentralized ride-sharing service. Thanks to the blockchain technology, no single entity or authority may monopolize the system for its own benefit or even to some drivers than others. Moreover, the selection process of drivers and riders works as a distributed auction that is handled over the blockchain which ensures transparency for both parties. Smart contracts are immutable and tamper-proof, and thus no party can alter their code or interfere with their execution without the consent of all the nodes in the blockchain network.

Privacy-preserving ride sharing activities. In our scheme, the privacy of riders and drivers (their ride requests/offers including the pick-up, drop-off and departure times) is protected by (i) replacing the rider's real identities by some placeholders (pseudonyms) for ride requests that correspond to temporary public-private key pairs. The pseudonym expires upon finishing a trip which ensures *unlinkabilty*. (ii) The use of generalization/cloaking technique in the bidding and selection phase, preserves the rider/driver trip information. Only the encrypted offers are stored on the public blockchain and no other sensitive information are leaked to the public. Additionally, as the size of the generalized area increases, more protection regarding rider's privacy can be achieved.

Achieving transparency. Since each user has an access to the blockchain, both drivers/riders can ensure that that their ride-requests or offers has been received and verified by the blockchain network. The bidding and selection phase is done on public and all driver can ensure their bids has been sent and received by the blockchain network. Thus, the proposed organization scheme offers high transparency which is not offered in case of using centralized approaches.

V. Related Work

Ride-sharing services have received a lot of attention in the literature as technology transforming modern societies [15].

Some companies have started developing a blockchain based ride sharing platform e.g., DACSEE [16] and Arcade City [17], but without drivers/riders privacy or anonymity. Existing ride-sharing systems can be categorized into centralized and distributed. In a centralized-setting, Xi *et al.* [18] adopts the Private Information Retrieval (PIR) technique and additive homomorphic encryption techniques to ensure privacy preserving shortest path computation. However, the computational overhead is quite high. Meanwhile [19] focused on matching between drivers and riders in ride-sharing systems while protecting the privacy of users against the untrusted service provider. The service provider uses a filtering protocol based on homomorphic arithmetic secret sharing and secure two-party equality test to decide the potential subset of drivers with whom the rider can travel. However, these schemes suffer from the inherent drawbacks of the client-server model and more importantly lacks transparency provided by our blockchain based scheme.

In [20], a general blockchain-based intelligent transportation framework has been proposed to mitigate the client-server model. Also, a case study is presented to show the impact of using blockchain in real-time ride-sharing services. Semenko *et al.* [21] have proposed a distributed platform for ride-sharing services. They suggest having an overlay network that includes all ride-sharing agencies called service nodes to constitute the network layer. The service node is responsible for matching drivers with riders. However, the platform needs trusted infrastructure to run on, and hence may fail in case of one service node is compromised which in turn will lead to inherent problems in the client-server model. Moreover, and unlike having blockchain of permissioned parties, our scheme is proposed atop the public blockchain which the lets any interested party to participate and leave. In more of a general context, blockchain technology has allowed progress in other areas outside of finance [22]–[24].

VI. Conclusion and Future Work

In this paper, we proposed a decentralized ride sharing organization scheme based on the revolutionary public blockchain. The scheme is decentralized since all interactions made by drivers/riders are done through the blockchain, and it does not need any company or organization to manage it. Moreover, there is no need to reveal any private information such a email address, phone number or credit card number. Using cloaked technique, only selected driver/rider know the exact pick-up location with out revealing any sensitive information to the public. Our evaluations shows that our scheme is practical with acceptable gas consumption and momentary cost. In the future work, we will design a privacy-preserving organization scheme considering the trust levels of riders and drivers in the RSSs.

VII. Acknowledgement

This research work was financially supported in part by NSF grant 1618549. In addition, parts of this paper, specifically Sections 1, 2 and 3, were made possible by NPRP grants NPRP10-1223-160045 from the Qatar National Research Fund (a member of Qatar Foundation). The statements made herein are solely the responsibility of the authors

References

[1] D. Schrank, B. Eisele, and T. Lomax, "2014 urban mobility report: powered by inrix traffic data," Tech. Rep., 2015.

[2] Ride sharing market cap. [Online]. Available: https://www.globenewswire.com/news-release/2019/01/17/1701096/0/en/218-Billion-Ride-Sharing-Market-Global-Forecast-to-2025.html

[3] New york times: Uber settles data breach investigation for $148 million. [Online]. Available: https://www.nytimes.com/2018/09/26/technology/uber-data-breach.html

[4] Motherboard: Uber china statement on service outage. [Online]. Available: https://motherboard.vice.com/en_us/article/3daa55/ubers-china-problem

[5] M. Baza, M. Nabil, N. Lasla, K. Fidan, M. Mahmoud, and M. Abdallah, "Blockchain-based firmware update scheme tailored for autonomous vehicles," *Proc. of the IEEE Wireless Communications and Networking Conference (WCNC), Marrakech, Morocco*, April 2019.

[6] M. Baza, M. Nabil, N. Bewermeier, K. Fidan, M. Mahmoud, and M. Abdallah, "Detecting sybil attacks using proofs of work and location in vanets," *arXiv preprint arXiv:1904.05845*, 2019.

[7] M. Baza, M. Nabil, M. Ismail, M. Mahmoud, E. Serpedin, and M. Rahman, "Blockchain-based charging coordination mechanism for smart grid energy storage units," *Proc. Of IEEE International Conference on Blockchain, Atlanta, USA*, July 2019.

[8] A. Kosba, A. Miller, E. Shi, Z. Wen, and C. Papamanthou, "Hawk: The blockchain model of cryptography and privacy-preserving smart contracts," *Proc. of the 2016 IEEE symposium on security and privacy (SP), California, USA*, pp. 839–858, 2016.

[9] Binomial coefficient. [Online]. Available: https://en.wikipedia.org/wiki/Binomial_coefficient

[10] F. Knirsch, A. Unterweger, and D. Engel, "Privacy-preserving blockchain-based electric vehicle charging with dynamic tariff decisions," *Computer Science-Research and Development*, vol. 33, no. 1-2, pp. 71–79, 2018.

[11] G. Wood, "Ethereum: A secure decentralised generalised transaction ledger," *Ethereum project yellow paper*, vol. 151, pp. 1–32, 2014.

[12] M. Baza, N. Lasla, M. Mahmoud, and M. Abdallah, "B-ride: Ride sharing with privacy-preservation, trust and fair payment atop public blockchain," *arXiv preprint arXiv:1906.09968*, 2019.

[13] Kovan etherum test net. [Online]. Available: https://kovan.etherscan.io

[14] Etherum gas station, "Available: https://ethgasstation.info/."

[15] T. V. A. Pham, I. I. Dacosta Petrocelli, G. F. M. Endignoux, J. R. Troncoso-Pastoriza, K. Huguenin, and J.-P. Hubaux, "Oride: A privacy-preserving yet accountable ride-hailing service," *Proc. of the 26th USENIX Security Symposium, BC, CANADA*, 2017.

[16] Dacsee platform. [Online]. Available: https://dacsee.com/

[17] Arcade city. [Online]. Available: https://arcade.city/

[18] Y. Xi, L. Schwiebert, and W. Shi, "Privacy preserving shortest path routing with an application to navigation," *Pervasive & Mobile Computing*, vol. 13, no. 4, pp. 142–149, June 2014.

[19] U. M. Aïvodji, K. Huguenin, M.-J. Huguet, and M.-O. Killijian, "Sride: A privacy-preserving ridesharing system," *Proc. of the 11th ACM Conference on Security & Privacy in Wireless and Mobile Networks*, pp. 40–50, 2018.

[20] Y. Yuan and F.-Y. Wang, "Towards blockchain-based intelligent transportation systems," *in Proc. of the IEEE 19th International Conference on Intelligent Transportation Systems (ITSC), Rio de Janeiro, Brazil*, pp. 2663–2668, Nov. 2016.

[21] Y. Semenko and D. Saucez, "Distributed privacy preserving platform for ridesharing services," Ph.D. dissertation, Inria-Sophia Antipolis, 2019.

[22] A. D. Dwivedi, L. Malina, P. Dzurenda, and G. Srivastava, "Optimized blockchain model for internet of things based healthcare applications," in *2019 42nd International Conference on Telecommunications and Signal Processing (TSP)*. IEEE, 2019, pp. 135–139.

[23] G. Srivastava, R. M. Parizi, A. Dehghantanha, and K.-K. R. Choo, "Data sharing and privacy for patient iot devices using blockchain," in *International Conference on Smart City and Informatization*. Springer, 2019, pp. 334–348.

[24] G. Srivastava, S. Dhar, A. D. Dwivedi, and J. Crichigno, "Blockchain education," in *2019 IEEE Canadian Conference of Electrical and Computer Engineering (CCECE)*. IEEE, 2019, pp. 1–5.