Cryptotransport: Blockchain-Powered Ride Hailing While Preserving Privacy, Pseudonymity and Trust

Conference Paper · November 2018

DOI: 10.1145/3274895.3274986

Yaron Kanza
AT&T Labs - Research

Eli Safra
Technion - Israel Institute of Technology

Cryptotransport: Blockchain-Powered Ride Hailing While Preserving Privacy, Pseudonymity and Trust

Yaron Kanza
AT&T Labs-Research
kanza@research.att.com

Eliyahu Safra
Technion–Israel Institute of Technology
safraeli@gmail.com

ABSTRACT

A *ride-hailing* service, where passengers choose the origin and destination of their ride, preserves *location privacy* if the origin and destination of the ride are not revealed to anyone other than the driver. *Pseudonymity* means that the driver and the passenger use a pseudonym and do not reveal their true identity to each other. The service is *trusted* if a dishonest passenger cannot use the service without paying, and a driver must provide the service to get paid. In this paper, we introduce *cryptotrasnport*—a ride-hailing service powered by cryptocurrency and blockchain—and we show that it can preserve privacy, pseudonymity, and trust, simultaneously. Furthermore, by using blockchain technology, cryptotransport matches riders with vehicles in a decentralized fashion, without relying on any company or organization to manage the system.

CCS CONCEPTS

•Information systems →Spatial-temporal systems; •Security and privacy →*Distributed systems security;*

KEYWORDS

Blockchain, privacy, anonymity, trust, ride hailing, cryptocurrency

ACM Reference format:
Yaron Kanza and Eliyahu Safra. 2018. Cryptotransport: Blockchain-Powered Ride Hailing While Preserving Privacy, Pseudonymity and Trust. In *Proceedings of 26th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, Seattle, WA, USA, November 6–9, 2018 (SIGSPATIAL '18),* 5 pages.
DOI: 10.1145/3274895.3274986

1 INTRODUCTION

Ride hailing—where people choose the origin and destination of their ride—is rapidly evolving with the vast growth in the popularity of the *sharing economy*, and the rise of *transportation network companies* like Uber, Lyft and DiDi. However, these companies are collecting a substantial amount of private information about their users (both passengers and drivers), including the origin and destination of the ride, user identities and various details about the ride, e.g., Uber specifies in https://privacy.uber.com/policy/ that

> *"If you are a rider, Uber may collect location information when the Uber app is running in the foreground. In certain regions, Uber may also collect this information when the Uber app is running in the background of your device if this collection is enabled through your app settings or device permissions."*

Users of these services are forced to share with the service providers sensitive private location information.

In a ride-hailing service, preserving *location privacy* means that the origin and destination of the ride (or any other location related to the ride) are not revealed to anyone other than the driver. Clearly, this is not the case when using ride services of transportation network companies.

Using an ordinary taxicabs and paying in cash makes it difficult to trace the ride, however, this is a limited solution because in many places it is impractical to wait for a coincidentally bypassing available taxicab. The ride should be called for, and personal or geospatial details may be collected in that process, e.g., the phone number that was used to place the order, the location from which the order was made and the pick-up location. Furthermore, some of the data could be released and de-anonymized, e.g., see in [12] an example of applying de-anonymization (based on [11]) to the NYC Taxi and Limousine Commission (TLC) data set. Therefore, ordinary taxicabs do not guarantee location privacy. More importantly, this solution is untrusted—if the passenger pays at the beginning of the ride, there is no guarantee that the driver will take the passenger all the way to the desired destination. If the passenger is supposed to pay at the end of the ride, then the driver is taking a risk of not getting paid when completing the ride. When the payment is through an intermediary service provider, like Uber, Lyft or DiDi, the payment system is commonly trusted, but this requires sharing sensitive information with the service provider.

In some cases, trust could be gained by letting the driver and rider know the identity of one another, because they may be reluctant to act in a way that could damage their reputation. However, some riders (drivers) may not want to expose their identity to their drivers (riders). The system provides *anonymity* if the driver and the passenger do not need to reveal their true identity to one another.

Is it possible to create a ride-hailing system in which location privacy, anonymity and trust are provided simultaneously? In layman terms the goal is to allow passengers to anonymously order a ride from an anonymous driver where the origin and destination of the ride are revealed only to the driver and such that the payment is according to the provided service, i.e., preventing fraudulent behavior of either drivers or passengers.

In this paper we present a *cryptotransport system*—a blockchain-based ride-hailing service that preserves location privacy, anonymity and trust. Blockchain is a transparent, tamper-proof and decentralized ledger that was initially proposed as a solution to the double-spending problem in cryptocurrencies [10]. Recently, many blockchain-powered applications were developed in various domains [6, 15, 16]. A decentralized ride-hailing system has several

SIGSPATIAL '18, Seattle, WA, USA
© 2018 Copyright held by the owner/author(s). 978-1-4503-5889-7/18/11...$15.00
DOI: 10.1145/3274895.3274986

advantages over a centralized one. First, there is no company that owns the data, has control over the information or that can revoke access to the service from particular users. Second, there is no single point of failure because the system is managed by many peers, and none of them has a central role in the system. Third, companies that do not trust one another could cooperate through the system, e.g., a company could serve customers using vehicles of its competitors while staying anonymous.

2 BACKGROUND

Blockchain is an immutable storage of transactions in a chain of blocks. The chain is created in a decentralized fashion by peers, using a peer-to-peer network, without any central entity to govern it or enforce rules. The chain structure provides a serialization of the stored transactions, to decide which transactions are valid.

Cryptocurrencies. In cryptocurrencies like Bitcoin, a transaction can be a reward to the creator of a block, or a transfer of coins from the owner to a payee. User identities are not revealed, to provide *pseudonymity*—the identity of the user is disguised and replaced by a pseudonym. Using a public key infrastructure, users create and maintain pairs of private and public keys. The public key is a pseudonym (or *address*) of the user, e.g., a transaction $t = (x \longrightarrow y, c)$ transfers c coins from address x to address y. The address y is the public key of the recipient, and only the user with the matching private key can spend the received coins.

Blockchain. To prevent double spending, the transactions are added to the blockchain. The chain defines a serialization of the transactions, so that if two transactions transfer the same coins (double spending), after the insertion of one of them into the blockchain, the other transaction is considered invalid, and should not be added to the blockchain. The blockchain, thus, represents a consensus of the peers on what are valid transactions.

The transactions are organized into blocks, which are created and added to the blockchain by members of a peer-to-peer network. In Bitcoin, these peers are called *miners*. The first block in the chain is the *genesis block*. Every other block contains a hash of the previous block in the chain, e.g., using SHA-256. This means that a change in one of the blocks would either result in an incorrect chain or require changing the hash values in all the following blocks.

A blockchain is tamper-proof, where changes of past blocks are practically impossible. To achieve that and to prevent forks, where a separation of the chain cannot be resolved, blockchains like Bitcoin rely on *proof-of-work* (PoW)—a computation that is hard and time consuming, e.g., a cryptographic riddle. In Bitcoin, each block includes a *nonce* such that the hash of the block (with the nonce) has at least k leading zeros. Computing the nonce is hard, hence it is a PoW. The value k is determined such that the overall computation by all the peers (miners) would require approximately 10 minutes for computing a block. In a case of a conflict, or a fork, miners are expected to add blocks to the longest valid branch. This causes short branches or branches that contain invalid blocks (e.g., double spending) to be abandoned and not be part of the chain. Eventually, this leads to consensus.

An attacker that tries to change a block in the blockchain needs to create an alternative branch and compete with all the other miners, in an attempt to make the alternative branch the longest one. The chances of succeeding are slim, due to the hardness of block creation. This provides immutability, stability and reliability.

While *permissionless* blockchains like Bitcoin are slow, *permissioned* blockchains, in which the peers are predefined, can provide much better performances, e.g., Hyperledger (https://www.hyperledger.org/) has a rate of more than 3500 transactions per second and a latency of less than one second. This could support a ride-hailing service in a large city, and a geospatial hierarchy of chains, as suggested in [6], could be used to extend the service to large areas. See a survey of blockchain technologies in [1].

Smart contracts. A *smart contract* is a code that is stored on the blockchain and executed by the peers that manage the blockchain, when particular conditions are met. A smart contract can be regarded as a self executing contract and it is intended to digitally facilitate, verify, and enforce the execution of a contract, without the need for an intermediary or a third party. Like other transactions on the blockchain, transactions that are a result of applying a smart contract are irreversible and visible to all the peers.

Ride sharing. Ride-sharing received a lot of attention in the literature [2–4, 8, 9]. In [7], a privacy-aware setting of ride-sharing was studied but it is based on obfuscation and is very different from our study. Centralized privacy-preserving ride hailing services are discussed in [13, 14]. Some companies develop a blockchain-based ridesharing platform, e.g., DACSEE (https://dacsee.io/) and Arcade City (https://arcade.city/), but without location privacy or anonymity. Using blockchain to allocate airspace for drones has been studied in [5]. To the best of our knowledge, the problem of using blockchain to provide location privacy, anonymity and trust in ride-hailing services has not been studied so far.

3 PROBLEM DEFINITION

We present now our model. We assume a model where the passenger specifies how much she is willing to pay when issuing a ride request. A *ride request* $req = (p, t_{orig}, l_{orig}, l_{dest}, fare)$ is a 5-tuple where p is the passenger, t_{orig} is the start time of the ride, l_{orig} and l_{dest} are the origin and destination, i.e., pick-up and drop-off locations of the intended ride. The *fare* is the offered payment. A ride $r(req) = (req, d)$ is a matching of the ride request *req* to a driver d.

Pseudonymity. As in Bitcoin, the user's identity is replaced by some placeholder (a pseudonym), which is the public key in a pair of public and private keys (e.g., using ECDSA or RSA).

Each user (driver or passenger) has matching pairs of public key K_{pub} and private key K_{priv}. The public key is used as a user pseudonym (also referred to as *address*). The user keeps the private key secret and can use it to prove ownership of the public key, i.e., association with the address. A message m can be signed using the private key $m' = sign(m, K_{priv})$. A verifier can then apply the public key $verify(m', K_{pub})$ to the signed message, to verify that m is associated with the owner of the address K_{pub}. Each user may have as many pairs of public and private keys as they need or desire.

To achieve pseudonymity, passengers and drivers (in ride requests and rides) are represented by an address (public key).

Location privacy. To preserve privacy, the passenger should only reveal the general area of the ride, without sharing the precise origin and destination with anyone other than the driver. Formally, a *privacy-preserving ride request* is a tuple $(K_{pub}, t_{orig}, A, dist, fare)$

where K_{pub} is the public key of the passenger, t_{orig} is the start time of the ride, A is the ride area, e.g., a city. The area A should be large enough to obfuscate the exact origin and destination. It should cover l_{orig} (covering l_{dest} may not be required). The value *dist* is an estimated travel distance, and *fare* is the offered fare of the ride. Note that *dist* is not necessarily the shortest distance between the pick-up and drop-off locations, that is $dist \geq \left\| l_{orig} - l_{dist} \right\|$. Accordingly, only the area of the ride and an upper bound on the distance between the origin and destination are revealed.

Trust. A trusted system should guarantee payment to the driver according to the actual transportation service that was provided. Reliability should be guaranteed regardless of the trustworthiness of the passenger and driver, since they remain pseudonymous. This is somewhat similar to exchange of Bitcoins—the payer, payee and miners are pseudonymous, without knowing whether any of them can be trusted, and yet, the system is reliable.

Cryptotransport. We refer to the concept of a blockchain-based ride hailing service, where the payments are by using a cryptocurrency, as *cryptotransport*. Ride requests and payments are all recorded on the blockchain in a decentralized way. The challenge is to provide pseudonymity, location privacy and trust, simultaneously. In the following section we explain how to do so.

4 CRYPTOTRANSPORT SYSTEM

We present now the cryptotransport system. To explain design choices, we start by presenting partial solutions, discuss their limitations and show how to cope with these limitations.

Publishing a ride request. A passenger p can publish a ride request as a self-transaction ($p \longrightarrow p, fare$), from p to p, with the fare of the ride in the transaction, to provide an indication that she has enough resources to pay for the ride. In a naive solution, the passenger would publish the entire request *req* on the blockchain and will wait for a driver to arrive. Note that there is no intermediary entity to coordinate the matching of a request with a driver.

This is improper for several reasons. First, it lacks location privacy, because the origin and destination of the ride are revealed. Second, there is no payment guarantee, so this solution is untrusted. Third, the passenger does not know if there is any driver who is willing to provide the service. Also, several drivers may arrive at the pick-up location simultaneously, which is problematic, and may cause a breach of privacy, by revealing drivers to one another. Hence, there should be a way to select the driver without relying on an intermediary. However, without publishing the pick-up location how will the driver and the passenger meet to start the ride?

Driver selection. Driver selection is as follows. First, the passenger p publishes a ride request as a self-transaction, with the following details: address of p, area of travel, travel distance and suggested fare. Any driver that is interested in the ride would try to add to the blockchain a self *ride-offer transaction* relating to the ride request. Like in the prevention of double spending, miners should avoid adding to the blockchain two ride-offer transactions for the same ride request. The added ride-offer transaction will determine the driver, and it should include the driver's address (public key). Then, the passenger would add a transaction in which the pick-up location is encrypted using the public key of the driver, so only the selected driver could decrypt it and know the pick-up location.

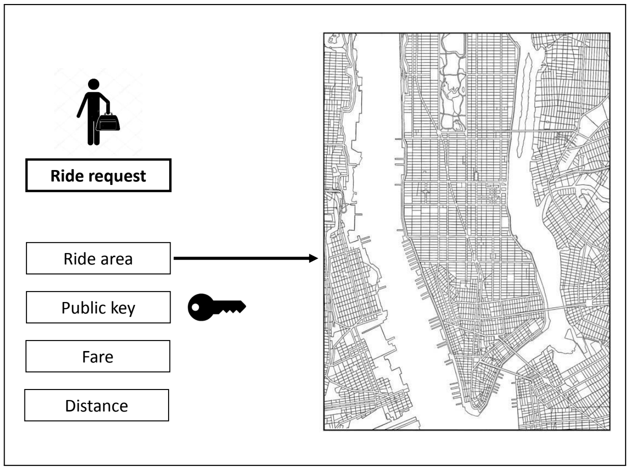


Figure 1: A ride request, without revealing identities (pseudonymity) or pick-up location (privacy).

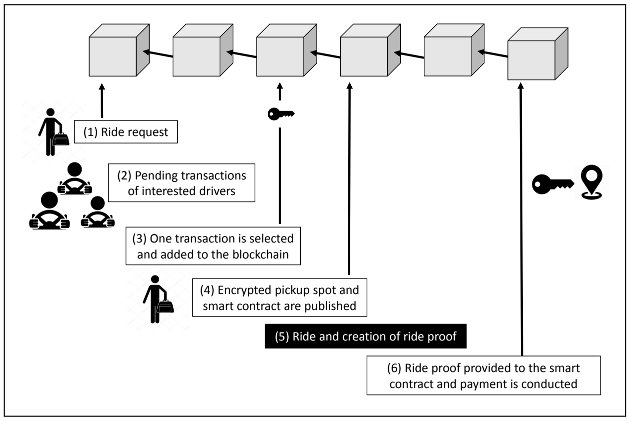

(1) Ride request
(2) Pending transactions of interested drivers
(3) One transaction is selected and added to the blockchain
(4) Encrypted pickup spot and smart contract are published
(5) Ride and creation of ride proof
(6) Ride proof provided to the smart contract and payment is conducted

Figure 2: The process of requesting a ride, getting a driver and paying, all via the blockchain.

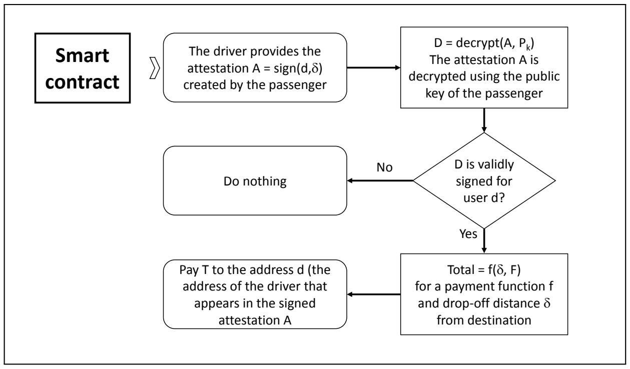

Smart contract
The driver provides the attestation A = sign(d,δ) created by the passenger
D = decrypt(A, P_k) The attestation A is decrypted using the public key of the passenger
D is validly signed for user d?
No
Do nothing
Yes
Total = f(δ, F) for a payment function f and drop-off distance δ from destination
Pay T to the address d (the address of the driver that appears in the signed attestation A

Figure 3: A smart contract created for public key (address) P_k, fare F and payment function f.

In cryptocurrencies each transaction has a fee that is given to the creator of the block containing the transaction. In the cryptotransport system, the transaction fee of a ride-offer has two important roles. First, drivers who offer a high transaction fee increase the chances of their transaction to be selected. This helps selecting drivers who are more sincere about their willingness to conduct the ride. Second, drivers will not 'bid' on rides they do not intend to perform to avoid paying a fee for nothing.

The driver selection is as follows.

(1) The passenger p publishes a privacy-preserving ride request, as a self transaction that includes the fare and the public key of p.
(2) Interested drivers try to add a ride-offer self-transaction associated with the ride request. Only one transaction can be selected and added to the blockchain. The added transaction contains the public key of the driver.
(3) Another self transaction of p is added to the blockchain where the pick-up location is encrypted using the public key of the selected driver.
(4) After decrypting the pick-up location, the driver can pick up the passenger from the agreed-upon location.

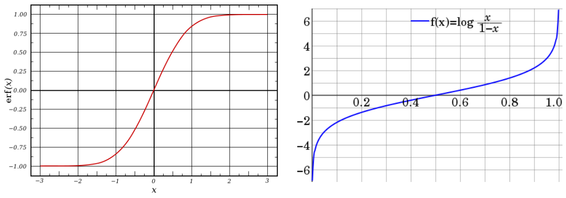
Note that the pick-up location is not revealed to anyone other than the driver, and the driver selection is conducted without any intermediary.

Trust. While the driver selection procedure provides location privacy and pseudonymity, its main limitation is that there is no payment warranty. How should the system protect passengers from dishonest drivers and protect drivers from dishonest riders?

Our solution is based on a smart contract. The smart contract is an algorithm that pays an allocated sum to a given address when particular conditions are met, i.e., when the driver provides a proof that the passenger reached the destination. This, however, still does not solve the problem. If the driver can create a proof without the involvement of the passenger, then the driver could fake a proof. If the driver needs the signature of the passenger at the end of the ride, then the passenger could act dishonestly and not provide the proof, even if the service has been fully provided.

To tackle this issue, let us start with the case where the drop-off location l_{dest} is visible in the ride request. In such a case, the smart contract would require a proof that contains the drop-off location and address of the driver signed by the private key of the passenger who created the contract. (The smart contract includes the public key of the passenger p, to verify the authenticity of the signature, and the address of the driver to prevent spending it on someone else.) If the signed location is near the destination, then the money will be delivered to the address of the driver—the one that appears in the smart contract and in the signed proof provided by the passenger. See illustration in Fig. 3.

The signed pair of location and driver key could be created using a device with a GPS, e.g., a smartphone, and sent to the driver. The driver will provide to the smart contract the attestation of driving the passenger to the destination (the signed pair). To prevent passengers from withholding the proof (and the payment), a proof with the current location can be created every short period of time, say every minute, and sent to the driver. The driver will upload the proof with the location that is closest to the destination, and the payment will be according to the distance of the uploaded location from the destination. In such a case, if the passenger stops generating proofs, the driver may stop the ride, and vice versa—if the driver stops at a point that is far from the drop-off location, the reward will be accordingly.



Error function

Logit

Figure 4: Potential payment functions.

The smart contract will be stored on the blockchain with the encrypted pick-up location. Since the blockchain is tamper-proof, the driver knows that the smart contract will be executed and the promised fare will be delivered if the suitable proof will be provided.

Different payment methods can be used for partial rides. The following are three examples.

- For a ride of fraction $0 \leq x \leq 1$ of the distance, the driver will get a portion of x^{α} of the fee and the rider will pay a portion of $1 - (1 - x)^{\alpha}$, where $\alpha \geq 1$. For $\alpha = 1$, the payment is a linear function. Otherwise, the difference between what is paid and what is received would be a transaction fee. A large α encourages both the driver and the rider to complete the ride, because the rider will pay a large portion of the fare already at the beginning of the ride and the driver will get a large portion only at the end of the ride.
- Logit: $f(x) = \log \frac{x}{1-x}$, for $0 < x < 1$ specifying the fraction of distance traveled. In this approach a large part of the fare is paid for the initial travel (to motivate the passenger to pay for the rest of the ride) and at the end (to motivate the driver to reach the end). See Fig. 4.
- Error function: $erf(x) = \frac{1}{\sqrt{\pi}} \int_0^x e^{-t^2} dt$, where there is a smaller payment at the beginning and at the end and most of the payment is at the middle of the ride, to encourage drivers and passengers to conduct at least most of the ride. See Fig. 4.

4.1 Combining location privacy and trust

The model in which the passenger reveals the drop-off location does not provide location privacy. A simple solution to this is as follows. The passenger will refrain from publishing the drop-off location. The smart contract will expect as a proof a signed pair of driver address and distance from destination, signed by the private key of the passenger (whose matching public key is in the smart contract). Now, during the ride, the passenger application will periodically generate signed distances, i.e., the distance between the current location and the destination. If the passenger stops producing signed distances, the driver may stop the ride. Note that signed distances can be exchanged between smartphones directly, using Bluetooth. The payment will be according to the signed distance provided to the smart contract by the driver. Note that this is done without revealing the pick-up and drop-off locations of the passenger to anyone other than the driver.

Let C be the presented cryptotransport system.

Proposition 4.1. *System C only reveals the area A in which the travel is conducted and an upper bound on the distance between the source and destination. Besides that, it provides location privacy.*

Proof. (Sketch) A ride request only reveals A and the distance bound. Also the smart contract is defined as a function of the distance bound. The pick-up location is stored on the blockchain but encrypted using the private key of the driver. If the driver does not reveal this location or loses the key, and if the encryption is strong, then the location information of the passenger and driver is not revealed. Other than that, no location information is exchanged with other drivers, miners or other passengers. □

Proposition 4.2. *System C provides pseudonymity to the passengers and drivers.*

Proof. (Sketch) In the protocols, all the information exchange and the interactions are by using a public key as a pseudonym of the passenger or driver. □

Proposition 4.3. *System C is trustworthy, that is, it does not allow passengers to get a ride without paying, and does not let drivers get the fare without making the ride.*

Proof. (Sketch) Suppose that a passenger wants to get a ride without paying. The passenger must add the smart contract to the blockchain—otherwise the driver will not agree to perform the ride. The blockchain is tamper-proof and the money dedicated for the payment must be secured when the smart contract is established. Hence, the passenger can only try to prevent the driver from getting a valid attestation. But, the driver can easily test the validity of the attestation and can refuse continuing the ride without valid proofs from the passengers. Once the attestations are given, the payments from the smart contract are secured because the smart contract cannot be spent on any address different from that of the driver.

Suppose that the driver tries to get the money without making the ride. The fare can only be given via the smart contract and a proof is required. If the private key of the rider is secured, the driver will not be able to create the proof needed for the payment. □

5 CONCLUSION

In this paper we illustrate how by using blockchain, cryptocurrency and smart contracts a decentralized ride-hailing service that preserves location privacy and pseudonymity could be trusted. The system is decentralized, does not require any company or organization to manage it, and the matching of drivers with ride requests is based on the consensus algorithm of the blockchain. Furthermore, all the interactions between passengers and drivers are via the blockchain—the ride request, matching a driver with a ride, the payment. There is no need to reveal any private information like email address, phone number or credit card number, and locations are only revealed to those who participate in the ride.

REFERENCES

[1] Dinh Tien Tuan Anh, Meihui Zhang, Beng Chin Ooi, and Gang Chen. 2018. Untangling Blockchain: A Data Processing View of Blockchain Systems. *IEEE Transactions on Knowledge and Data Engineering* (2018).

[2] Mohammad Asghari, Dingxiong Deng, Cyrus Shahabi, Ugur Demiryurek, and Yaguang Li. 2016. Price-aware real-time ride-sharing at scale: an auction-based approach. In *Proc. of the 24th ACM SIGSPATIAL International Conference*. ACM.

[3] Bin Cao, Louai Alarabi, Mohamed F Mokbel, and Anas Basalamah. 2015. Sharek: A scalable dynamic ride sharing system. In *16th IEEE MDM*.

[4] Blerim Cici, Athina Markopoulou, and Nikolaos Laoutaris. 2015. Designing an on-line ride-sharing system. In *Proc. of the 23rd ACM SIGSPATIAL*.

[5] Tamraparni Dasu, Yaron Kanza, and Divesh Srivastava. 2018. Geofences in the Sky: Herding Drones with Blockchains and 5G. In *Proc. of the 26th ACM SIGSPATIAL International Conference*. ACM.

[6] Tamraparni Dasu, Yaron Kanza, and Divesh Srivastava. 2018. Unchain your blockchain. In *Proc. of the Symposium on Foundations and Applications of Blockchain (FAB'18)*. 16–23.

[7] Preeti Goel, Lars Kulik, and Kotagiri Ramamohanarao. 2016. Privacy-Aware Dynamic Ride Sharing. *ACM Trans. Spatial Algorithms Syst.* 2, 1 (2016).

[8] Shuo Ma and Ouri Wolfson. 2013. Analysis and evaluation of the slugging form of ridesharing. In *Proc. of the 21st ACM SIGSPATIAL*. ACM, 64–73.

[9] Shuo Ma, Yu Zheng, and Ouri Wolfson. 2015. Real-time city-scale taxi ridesharing. *IEEE Transactions on Knowledge and Data Engineering* 27, 7 (2015), 1782–1795.

[10] Satoshi Nakamoto. 2008. Bitcoin: A peer-to-peer electronic cash system. (2008).

[11] Arvind Narayanan and Vitaly Shmatikov. 2008. Robust de-anonymization of large sparse datasets. In *Symposium on Security and Privacy*. IEEE, 111–125.

[12] Vijay Pandurangan. 2014. On taxis and rainbows: Lessons from NYC's improperly anonymized taxi logs. https://tech.vijayp.ca/of-taxis-and-rainbows-f6bc289679a1. (2014).

[13] Thi Van Anh Pham et al. 2017. ORide: A Privacy-Preserving yet Accountable Ride-Hailing Service. In *Proceedings of the 26th USENIX Security Symposium*.

[14] Michael Rigby, Antonio Krüger, and Stephan Winter. 2013. An opportunistic client user interface to support centralized ride share planning. In *Proc. of the 21st ACM SIGSPATIAL International Conference*. ACM, 34–43.

[15] Melanie Swan. 2015. *Blockchain: Blueprint for a new economy.* " O'Reilly Media, Inc.", Sebastopol, CA, USA.

[16] Don Tapscott and Alex Tapscott. 2016. *Blockchain Revolution: How the Technology Behind Bitcoin Is Changing Money, Business, and the World.* Penguin Random House, New York, NY, USA.