Blockchain Based Secure Package Delivery via Ridesharing

Xuefei Zhang†, Junjie Liu†, Yijing Li†, Qimei Cui†, Xiaofeng Tao† and Ren Ping Liu§
†National Engineering Lab for Mobile Network Technologies,
Beijing University of Posts and Telecommunications, Beijing, China
§University of Technology Sydney, Sydney, Austrilia
Email: {zhangxuefei, JunjieLiu, liyijing, cuiqimei, taoxf}@bupt.edu.cn, RenPing.Liu@uts.edu.au

***Abstract*—Delivery service via ridesharing is a promising service to share travel costs and improve vehicle occupancy. Existing ridesharing systems require participating vehicles to periodically report individual private information (e.g., identity and location) to a central controller, which is a potential central point of failure, resulting in possible data leakage or tampering in case of controller break down or under attack. In this paper, we propose a Blockchain secured ridesharing delivery system, where the immutability and distributed architecture of the Blockchain can effectively prevent data tampering. However, such tamper-resistance property comes at the cost of a long confirmation delay caused by the consensus process. A Hash-oriented Practical Byzantine Fault Tolerance (PBFT) based consensus algorithm is proposed to improve the Blockchain efficiency and reduce the transaction confirmation delay from 10 minutes to 15 seconds. The Hash-oriented PBFT effectively avoids the double-spending attack and Sybil attack. Security analysis and simulation results demonstrate that the proposed Blockchain secured ridesharing delivery system offers strong security guarantees and satisfies the quality of delivery service in terms of confirmation delay and transaction throughput.**

***Keywords*—Blockchain, PBFT, Ridesharing, Security**

I. Introduction

Ridesharing provides partner matching services for requestors (e.g., passengers or packages) and providers (e.g., drivers of private cars or taxis) with similar or overlapping travel paths [1]. In ridesharing delivery, drivers offer peer-to-peer ridesharing trips with assured quality of delivery service for packages [2]. It is a cost-effective logistic channel, bring cost savings to package delivery for customers, as well as extra benefits for the participating drivers. Moreover, ridesharing also benefits the society by reducing traffic congestion and carbon emission [3].

Recently, ridesharing delivery service has attracted increased attention. For example, Liu *et al.* provided a private-car-assisted ridesharing delivery system with roadside delivery boxes, where a centralized matching method is adopted to improve the profits of drivers [4]. In addition to private cars, Ma *et al.* noticed that taxis with low occupancy can also be used for delivery [5]. Furthermore, Febbraro *et al.* designed a central matching platform to achieve hybrid ridesharing [6] between both passengers and package deliverie. The upper bound of delivery capacity is derived in [7] [8], which demonstrated the enormous potential of ridesharing delivery to improve traffic efficiency.

Corresponding Author:Xuefei Zhang

Existing ridesharing delivery systems are mostly based on centralized architectures, where the providers report their real-time location periodically to a cental server. However, such centralized architecture may bring about data leakage to threaten individual privacy. Moreover, the inherent defect of centralized ridesharing system, i.e., single-server storage and control for high-valued transaction data, results in a huge economic and credit damage once the data tampering happens. In order to solve these challenges, a Blockchain based ridesharing system has been proposed in this paper for the first time.

Blockchain is an open, distributed ledger that records transactions in a verifiable and permanent way, which is the underlying fabric for Bitcoin. Due to its great potential on security and trust, Blockchain technology has been applied in many areas to improve data security and trust. For instance, Liu *et al.* proposed a Blockchain-enabled data sharing scheme in Internet of Things [9]. By combining with deep reinforcement learning, the security and reliability of data sharing can be guaranteed. In [10], a self-organized wireless access network based on Blockchain was implemented, where the mobile devices can complete the network switching without the guidance from base stations. [11] achieved secure and self-organized data sharing in Internet of Vehicles by introducing smart contract technology. The authors of [12] expanded Blockchain and smart contract into medical systems to build a decentralized medical data sharing system.

Motivated by the above observations, we propose a Blockchain secured ridesharing delivery system with guaranteed quality of delivery service. The proposed Blockchain system has the advantages of a decentralized ridesharing system while ensuring data security and privacy. However, the long confirmation delay of public Blockchain is unbearable for ridesharing processes, therefore the Blockchain consensus algorithm needs to be improved. Moreover, security mechanisms need to be designed to defend against double-spending and Sybil attacks in the proposed Blockchain system. The contributions of this paper are summarized as follows:

- We propose a Blockchain secured ridesharing delivery system. To the best of our knowledge, it is the first distributed ridesharing system. By leveraging the Blockchain immutable and distributed architecture, the data leakage and data tampering threats of the ridesharing system are effectively addressed.

- A Hash-oriented PBFT consensus algorithm is designed to reduce the confirmation delay from 10 minutes to 15 seconds in the Blockchain-based system. Security analysis demonstrates that the proposed algorithm can prevent the common Blockchain attacks, including double-spending and Sybil attacks.
- Simulation results show that the proposed Blockchain secured ridesharing system offers strong security guarantees and satisfies the quality of delivery in terms of the transaction confirmation delay of 15s, and the system throughput of 15 transactions per second.

The rest of this paper is organized as follows. The system model and two attack models are introduced in Section II. Section III describes the overall flow of Blockchain secured Ridesharing Delivery system. In Section IV, simulation results illustrate the performance of the proposed system on delay and transaction throughput. Finally, we draw the conclusion in Section IV.

II. System Model

In this paper, we propose a Blockchain secured Ridesharing Delivery system as shown in Fig.1. This system involves four components: requestors, providers, attackers, and mobile edge computing (MEC) servers. The j^{th} requestor and the m provider is denoted by u_j and v_m for $1 \leq j \leq N$ and $1 \leq j \leq M$, where N and M is the number of requestors and providers in the system. Each requestor sends a logistics request $r_{id} \triangleq (o, d, t)$ to nearby providers for package delivery service, where id represents the identifier of a request, o represents the origin of the packages delivery, d represents the destination of the packages delivery, and t represents the submission time of the request. Notably, identifier id and the submission time t_{id} are automatically generated by the system. Each requestor has an asymmetric key pair and certificate assigned by the key distribution center for signature and identity authentication. All Blockchain-related operations (e.g., proof of work (PoW), consensus, etc) are carried out in MEC servers. Commonly, attackers are equipped with strong computation capability, posing a serious threat to data privacy and security. Two common attack models are described in the following.

A. Double Spending Attack

Double spending attack, also referred as alternative history attack in cryptocurrency, is a potential security loophole in the most of distributed systems [10]. An attacker equipped with powerful computing capability can prepare a Blockchain fork. Even the transaction has been confirmed by the current main chain, an attacker can still alter the transaction history by releasing the fraudulent fork. In the package delivery ridesharing system, the double spending attack may result in huge economic damage to both providers and requestors.

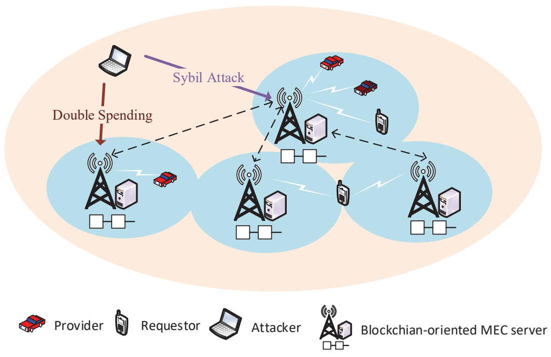

Fig. 1. System model of package delivery ridesharing system.

B. Sybil Attack

In a peer-to-peer network, a single node can masquerade as multiple nodes by acquiring multiple identities, therefore its influence on the voting results can be improved. In Practical Byzantine Fault Tolerance (PBFT) based Blockchain system, a new block can be successfully added to the main chain only after more than two thirds of the node votes to confirm it [13]. Therefore, by implementing the Sybil attack, the attacker alters the voting results by disguising themselves as multiple nodes to paralyze the system.

III. Blockchain Secured Ridesharing Delivery System

In this section, we provide a detailed description of the Blockchain secured Ridesharing Delivery system. First, we provide an overview of the transaction completion process. Second, the self-construction process of smart contract is presented. Third, we describe the new block generation to show how the Blockchain works in the proposed ridesharing system.

A. Overall Flow of Blockchain Based Ridesharing System

To clearly illustrate the overall process of the proposed ridesharing system, we consider an easy understanding example in the following, and the specific interaction steps between requestor and provider have been shown in Fig.2. The following steps are corresponding to the labels in the figure.

1) Request Broadcast: An requestor in the system generates a request involving the origin, the destination and the submission time of the package delivery. Noticeably, the origin and destination are the location of the mail box deposited packages, which avoid revealing the location of the package owner. Then, the request is broadcast to the nearby providers.

2) Smart Contract Confirmation: After receiving some requests, the provider performs the distributed matching to search a desirable package delivery request according to their requirements. If the provider decides to deliver the package, it creates a smart contract involving the description of the package and the delivery information, and then attaches its digital signature at the end.

3) Contract Broadcast: In the following, the smart contract is sent to the requestor for authentication, after which the status of the provider is locked and no other requests will be received. In this way, the requestor potentially receives more than one smart contract, but only one contract is permitted to create by the requestor according to their customized criteria, such as delivery time priority or cost priority. Finally, the requestor attaches the digital signature to the selected smart contract and broadcasts comfirmed contract to all MEC servers in the system.

4) New Block Generation: After receiving the confirmed smart contract, the MEC server checks the validity of the signatures of provider and requestor. Then, MEC servers aggregate all smart contact received in the previous period to create a new block, denoted by $newBlock$.

5) Hash-oriented Leader Selection: In this stage, MEC server generates a Merkle hash value based on its own identifier together with $newBlock$. Subsequently, the MEC server with the lowest hash value is selected as the leader.

6) Consensus over MEC servers: Once the leader has been determined, every MEC server verifies the content of the $newBlock$ sent by the leader, and attaches its digital signature at the end of the new block if the validity of $newBlock$ is confirmed. Then, replicas of the confirmed $newBlock$ are sent to other MEC servers. Meanwhile, every MEC server receives replicas from others. Until the received number of confirmed replicas exceeds two thirds of the total number of MEC servers, a confirmation message will be to the leader.

7) Execution: When the confirmation messages received by the leader exceeds two thirds of the total number of MEC servers, it indicates that the $newBlock$ has been successfully added to the Blockchain. Finally, the leader returns an acknowledgement message to both the requestor and provider, after which the provider will start to deliver the package with respect to the information in the smart contract.

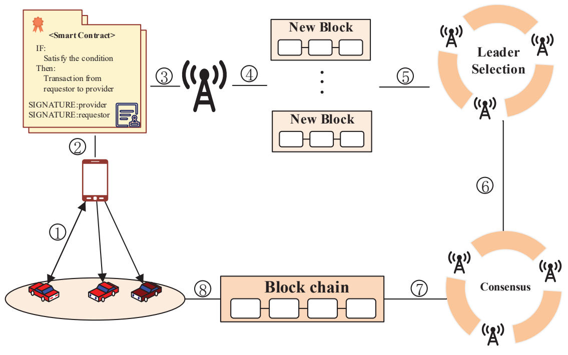

Fig. 2. Overall flow of the ridesharing system.

B. Smart Contract for Ridesharing Transaction

Smart contract, a standardized digital certificate to enable the transaction process to be self-organized, is utilized to record the ridesharing transaction in our system (a rough process is mentioned in Section III-A1 to A3). In this subsection, three steps for creating a smart contract for ridesharing transaction are provided in detail.

1) Request Generation: We consider that a requestor u_j produces a package delivery request r_{id}. The request format is given by

$$message_{id} = \left(id \left\| r_{id} \right\| Cert_{u_j} \left\| Sig_{u_j} \right.\right) \tag{1}$$

where $Cert_{u_j}$ is the certificate of u_j, Sig_{u_j} refers to digital signature generated by SK_{u_j}(the secret key of u_j)

$$Sig_{u_j} = Sign_{SK_{u_j}}\left(r_{id}\right) \tag{2}$$

Considering the origin of the request r_{id} and path planning suggestion, the provider decides whether to accept the request.

2) Smart Contract Confirmation: If the provider decides to deliver a packages, the provider will generate a smart contract according to request r_{id}. The format of the smart contract from provider v_m to requestor u_j is as follows.

$$v_m \rightarrow u_j : \text{SC=}\left(id \left\| r_{id} \right\| d_{id} \left\| C_{id} \right\| Sig_{v_m}\right) \tag{3}$$

where d_{id} is the service capability (e.g., pre-evaluated delivery time, cost, etc) offered by provider v_m and C_{id} is the license of the provider. After that, the provider v_m stops receiving package delivery request from requestors.

On the other side, it is possible that the requestor u_j receives a number of smart contracts, but only one contract is selected in light of its own customized criteria (e.g. delivery time priority, cost priority, etc).

3) Contract Broadcast: After the smart contract selection, the requestor adds its signature to the selected smart contract, which represents that the requestor and the provider have completed the ridesharing matching. Then, the contract is broadcast to all MEC severs in the system, as is shown in (4).

$$u_j \rightarrow MEC : \text{SC=}\left(id \left\| r_{id} \right\| d_{id} \left\| C_{id} \right\| Sig_{v_m} \left\| Sig_{u_j} \right.\right) \tag{4}$$

Then, every MEC server verifies the validity of the signature in the smart contract.

C. PBFT based Consensus Algorithm

A consensus algorithm in our system is a procedure through which all the MEC servers of the Blockchain network reach a common agreement about transactions in a newblock (a rough process is mentioned in Section III-A5 to A6). In this subsection, a detailed description of Hash-oriented PBFT consensus algorithm is given, and further we prove that this algorithm can effectively promote the efficiency of the consensus algorithm.

1) Traditional Consensus Algorithm: First, we introduce the basic idea of a traditional consensus mechanism, i.e., proof of work (PoW). A PoW is a piece of data which is difficult (costly, time-consuming) to produce but easy for others to verify and which satisfies certain requirements. Producing a PoW is a random process with low probability so that a lot of trial and error is required on average before a valid PoW is generated. The general form of workload proof is given by,

$$Hash\left(\text{Prev Hash} + newBlock + nonce\right) < \text{Given value} \tag{5}$$

where $nonce$ represents a random parameter, $newBlock$ is the new generated block composed of smart contracts, and Given value is a predefined value which can be used to adjust the computation difficulty level of Blockchain. The structure of a $newBlock$ is given by

Index	**Timestamp**
Previous Hash	**SelfHash**
Transaction Data	

Based on the irreversibility of the hash function, the only way to find a desirable $nonce$ is to conduct a great amount of attempts.

In the current public Blockchain, the computing time of PoW is around 10 minutes [14]. Moreover, in order to avoid the Blockchain fork, the final confirmation of a transaction needs to wait until the transaction has been added to a block and six blocks have been linked after it (about an hour) [14], which is unbearable for both requestor and provider in package delivery ridesharing. Meanwhile, PoW method poses a very high demand on the computing capability, i.e., the miner needs to continuously carry out the hash value calculation. Considering that the Blockchain is deployed in MEC servers in the proposed ridesharing system, it is unpalatable to occupy large computational power of public servers only for carrying out the mining activities of Blockchain.

2) Hash-oriented PBFT Consensus Algorithm: In order to tackle the long confirmation delay and demand of high computation capability mentioned above, we propose a Hash-oriented PBFT consensus algorithm as is shown in Fig.3.

First, the requestor sends a signed smart contract to all MEC servers during the broadcast period. After receiving the contract, $newblock$ is generated and proof of device(PoD) is conducted to select a leader based on (6) [10].

$$Hash\left(\text{Prev Hash} + newBlock + ID\right) < \text{Given value} \tag{6}$$

In the PoD-based mining, the hash value is produced in light of the fixed ID value of the device, instead of a great deal of tries in PoW. Thus, the number of hash calculation sharply drops to one attempt, thereby reducing the computational complexity. In addition, another merit of PoD is the lower computation difficulty of (6) merely by increasing Given value, thereby achieving lower confirmation delay. Among the hash values satisfying the Given value, the server with the lowest hash value is selected as the leader.

$$leader = \left\{ ID \left| \min\left(Hash\left(\text{Prev Hash} + newBlock + ID\right)\right)\right.\right\} \tag{7}$$

After the leader selection, all servers in the system verify the validity of the transaction written in the leader's $newblock$. Once the validity has been confirmed, servers add their digital signatures on the $newBlock$ and send them to other MEC servers in the system, as is shown in (8).

$$\begin{aligned} &MEC_i \to MEC_j : \\ &message_{MEC_i} = \left(leader \left\| newblock \right\| Sig_{MEC_i}\right) \end{aligned} \tag{8}$$

where MEC_i, MEC_j indicates the i^{th}, j^{th} MEC server respectively, $1 \le i, j \le K, i \ne j$, K is the number of MEC server in the system and

$$Sig_{MEC_i} = Sign_{SK_{MEC_i}}\left(leader \left\| newblock\right.\right) \tag{9}$$

In the following, the system performs the confirmation process in Fig.3 by exchanging message in (8) to verify the signature. When the number of valid received messages exceeds two thirds of the total number of servers, the server MEC_j will add the $newBlock$ to the end of the Blockchain and return a confirmation message to the leader.

$$\begin{aligned} &MEC_j \to leader : \\ &message_{conf} = \left(leader \left\| newblock \right\| Sig_{MEC_j}\right) \end{aligned} \tag{10}$$

Finally, in the execution process of Fig.3, the leader returns an acknowledgment to the requestor and the provider after receiving the confirmation message sent by more than two thirds servers, indicating that the transaction information has been successfully added to the Blockchain.

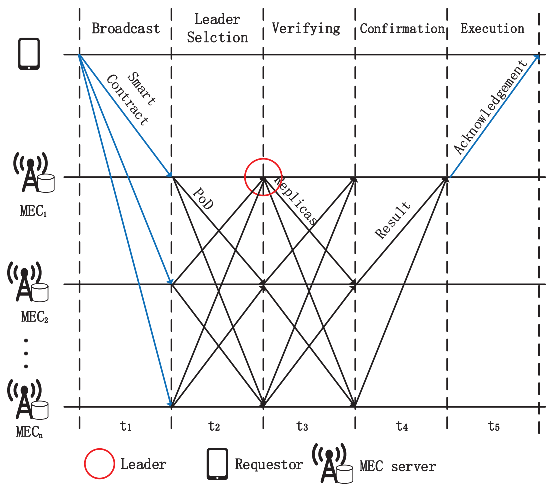

Fig. 3. The consensus process of Hash-oriented PBFT.

IV. Security Analysis

In this section, the security evaluation of the proposed Blockchain-based ridesharing system under the two classic attack models provided in II, are analyzed.

A. Double Spending Attack

Double spending attack utilizes Blockchain fork to alter the confirmed transaction records, resulting in huge economic damage to both requestors and providers. In the proposed package delivery ridesharing system, the Hash-oriented PBFT consensus algorithm is adopted to eliminate the Blockchain fork, the transaction needs to be verified by more than two thirds servers of the system before being successfully added to the Blockchain, thereby protecting the system from double spending attack.

B. *Sybil Attack*

Attackers launch Sybil attack by acquiring multiple identities, thereby improving its influence on the voting results. However, in the proposed ridesharing system, the identifier and asymmetric key pair, are bounded to a specific MEC server, which means it is impossible for a single server to obtain multiple identities. Meanwhile, attackers may generate fake asymmetric keys to masquerade as multiple servers. However, the validity of the digital signature generated by the asymmetric keys will be verified by all the MEC servers in the system, thereby preventing the system from being cheated.

V. Simulation Result and Discussion

In this section, we evaluate the performance of the proposed ridesharing system in terms of the varying number of participating MEC servers and block size. Moreover, the classic Bitcoin and Ethereum system are considered as the benchmark schemes.

A. *Confirmation Delay*

Confirmation delay is the time from the request initiation to the request is confirmed by the Blockchain system, which is a significant performance indicator for ridesharing transactions. In this subsection, we evaluate the confirmation delay of the ridesharing system in terms of the number of participating MEC servers from 100 to 1600 with the block size from 40 transactions/block to 160 transactions/block.

As is shown in Fig.4, the confirmation delay of a new block increases linearly with the growing number of MEC servers and block size. This is because every server receives more replicas as the number of MEC servers increases, thereby results in longer time to check the validity of the replicas. In addition, the larger block size brings about the longer confirmation time due to more transactions to be verified.

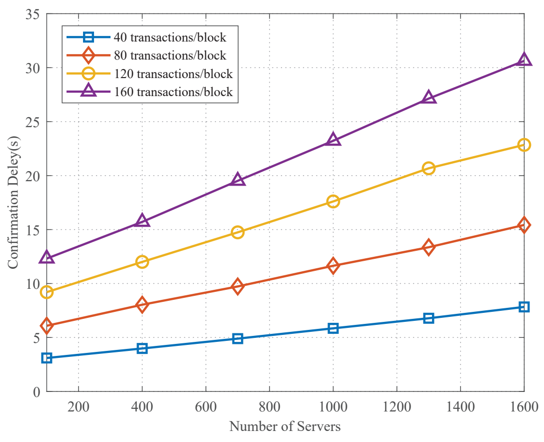

Fig. 4. Average confirmation delay with 0 to 1600 servers.

B. *CDF of Confirmation Delay*

In this subsection, we explore the cumulative probability distribution (CDF) of confirmation delay, which represents the completion probability under certain confirmation delay.

The curves in Fig.5 moves right along with the larger block size and more MEC servers, because the growth results in more transactions to be verified. Meanwhile, it is observed that the proposed ridesharing system can complete the confirmation process in 15s with a probability approximately to one hundred percent, under block size is 40 transactions/block and the number of MEC servers reaches to 1500. The average confirmation time of our proposed system is far less than that of Bitcoin system which is around 10 minutes.

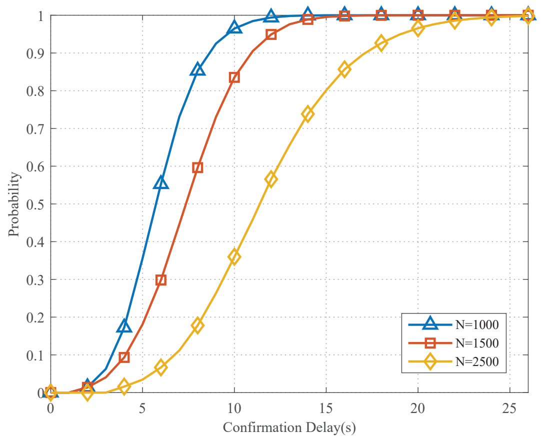

Fig. 5. CDF of confirmation delay in different number of servers and block size.

C. *Transaction Throughput*

Transaction throughput refers to the number of transactions that can be processed per unit time in the proposed ridesharing system. As shown in Fig.6, it is worth noting that the transaction throughput of the system grows to a stable value with the ascending block size. The reason is that the time cost of a single node is proportional to the block size, the expansion of block size means longer time should be taken to verify the transactions.

Fig.7 show the transaction throughput in terms of the the number of MEC servers. It can be seen that the advantage of the proposed system is obvious compared to Bitcoin and Ethereum system, but the increasing number of MEC servers brings about a sharp drop of transactions throughput. The cross point tells us that the proposed system is promising when the number of MEC server is less than 200. Considering the practical situation that the number of MEC server with a strong computation and storage capability to participate in ridesharing in a city is less than 200, the proposed system always performs best among the three systems and can satisfy the requirement of the transaction throughput of a online ridesharing application (At present, the transaction throughput of Uber, Lyft, etc., is about 12 transactions/s). Thus, Fig.6 and Fig. 7 provide a suggestion for the proposed ridesharing configuration.

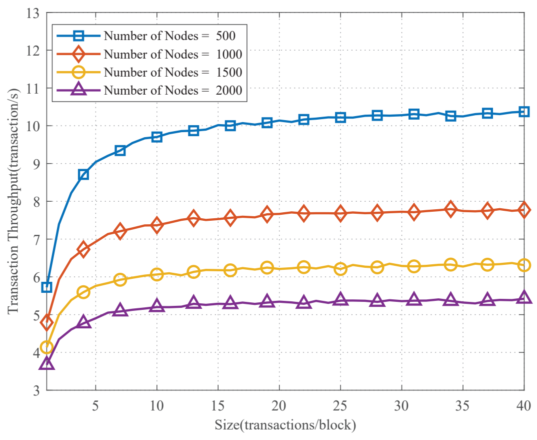

Fig. 6. Average transaction throughput with block size rise from 1 to 40 transactions.

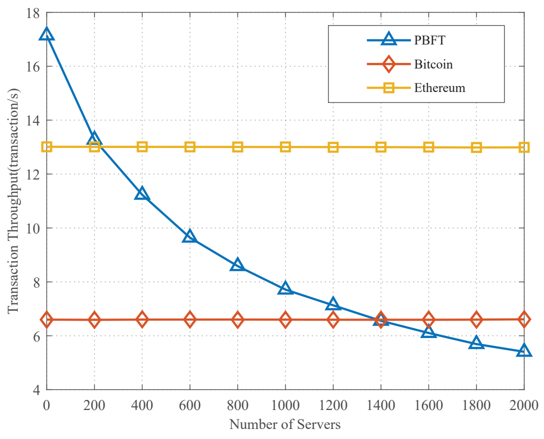

Fig. 7. Transaction Throughput comparison with Bitcoin and Ethereum in different number of servers.

VI. Conclusion

A Blockchain secured ridesharing delivery system is proposed for the first time to tackle the data leakage and tampering of the ridesharing system. Specifically, by leveraging the immutability and distributed architecture of Blockchain technology, we create smart contracts for ridesharing and propose a Hash-oriented PBFT consensus algorithm. The system is designed to reduce the confirmation delay and to avoid double-spending and Sybil attacks. Security analysis and simulation results demonstrate that the proposed package delivery ridesharing system offers strong security guarantees and satisfies the quality of delivery service interms of confirmation delay and transaction throughput.

Acknowledgment

This work was supported in part by the National Nature Science Foundation of China under Grant 61701037 and 61325006, in part by National Youth Top-notch Talent Support Program, and in part by the 111 Project of China B16006.

References

[1] Y. He, J. Ni, X. Wang, B. Niu, F. Li, and X. Shen, "Privacy-preserving partner selection for ride-sharing services," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 7, pp. 5994–6005, July 2018.

[2] M. Zhu, X. Liu, and X. Wang, "An online ride-sharing path-planning strategy for public vehicle systems," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 2, pp. 616–627, Feb 2019.

[3] M. Lindholm and S. Behrends, "Challenges in urban freight transport planning–a review in the baltic sea region," *Journal of Transport Geography*, vol. 22, pp. 129–136, 2012.

[4] X. Liu, Y. Liu, N. N. Xiong, N. Zhang, A. Liu, H. Shen, and C. Huang, "Construction of large-scale low-cost delivery infrastructure using vehicular networks," *IEEE Access*, vol. 6, pp. 21 482–21 497, 2018.

[5] S. Ma, Y. Zheng, and O. Wolfson, "Real-time city-scale taxi ridesharing," *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 7, pp. 1782–1795, July 2015.

[6] A. Di Febbraro, D. Giglio, and N. Sacco, "On exploiting ride-sharing and crowd-shipping schemes within the physical internet framework," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, Nov 2018, pp. 1493–1500.

[7] J.-F. Rougès and B. Montreuil, "Crowdsourcing delivery: New interconnected business models to reinvent delivery," in *1st international physical internet conference*, 2014, pp. 1–19.

[8] A. S. Oliveira, M. W. Savelsbergh, L. Veelenturf, and T. van Woensel, "Crowd-based city logistics," in *Sustainable Transportation and Smart Logistics*. Elsevier, 2019, pp. 381–400.

[9] C. H. Liu, Q. Lin, and S. Wen, "Blockchain-enabled data collection and sharing for industrial iot with deep reinforcement learning," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 6, pp. 3516–3526, June 2019.

[10] X. Ling, J. Wang, T. Bouchoucha, B. C. Levy, and Z. Ding, "Blockchain radio access network (b-ran): Towards decentralized secure radio access paradigm," *IEEE Access*, vol. 7, pp. 9714–9723, 2019.

[11] 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*, pp. 1–1, 2019.

[12] S. Wang, J. Wang, X. Wang, T. Qiu, Y. Yuan, L. Ouyang, Y. Guo, and F. Wang, "Blockchain-powered parallel healthcare systems based on the acp approach," *IEEE Transactions on Computational Social Systems*, vol. 5, no. 4, pp. 942–950, Dec 2018.

[13] M. Castro, B. Liskov *et al.*, "Practical byzantine fault tolerance," in *OSDI*, vol. 99, 1999, pp. 173–186.

[14] S. Nakamoto *et al.*, "Bitcoin: A peer-to-peer electronic cash system," 2008.