Hindawi
Mathematical Problems in Engineering
Volume 2022, Article ID 5040766, 7 pages
https://doi.org/10.1155/2022/5040766

Research Article

Privacy-Preserving Cross-Zone Ride-Matching for Online Ride-Hailing Service

Hui Ma, Yuan Ping, and Yong Zhang

School of Information Engineering, Xuchang University, Xuchang 461000, China

Correspondence should be addressed to Yuan Ping; pyuan.lhn@xcu.edu.cn

Received 14 April 2022; Revised 9 May 2022; Accepted 11 May 2022; Published 31 May 2022

Academic Editor: Wen-Tsao Pan

Although online ride-hailing supplies the nearest taxi matching for riders, the potential leakage of riders' hobbies and physical condition raises privacy concerns. Since most of the privacy-preserving schemes do limited matching of riders and drivers in the same zone, in this paper, we propose a novel privacy-preserving cross-zone ride-matching scheme, namely, Cride, which extends one zone into multiple neighboring zones. Based on the zone division of a city, CRide allows distance computation between rider and driver across adjacent zones in the encrypted domain. Furthermore, towards efficiency improvement, a ciphertext packing technique is introduced. Theoretical analysis and experimental results suggest that CRide achieves a high ride-matching accuracy and acceptable efficiency without leaking privacy.

1. Introduction

Today, online ride-hailing (ORH) services like DiDi and Uber have been welcomed by more and more people who use it to resolve the problem of taking a taxi at rush hour [1, 2]. Compared with traditional taxi services, ORH provides people with convenient services. Riders can request a driver just using his mobile phone in a short time, instead of standing by the street and waiting for a taxi.

Despite the advantage of ORH, the way of traveling brings challenges of privacy leakage [3, 4]. To offer ride-matching, ORH server (*RS*) needs to collect riders' and drivers' sensitive information, such as identities, locations, and traveling time, which is used to implement matching between drivers and riders. The ride-hailing data may be leaked and used to monitor the traces of riders, infer the hobbies and physical condition of riders, etc. Riders will become the target of junk mail, robbery, or blackmail attacks [5, 6]. Hence, it is essential to protect the privacy of riders and drivers.

In ORH service, to protect the sensitive data about drivers and riders, some works [7, 8] studied the privacy-preserving ORH service. PrivateRide [7] was proposed to protect the privacy of ride-hailing. Later, ORide [8] was designed to find the nearest driver for the rider without leaking identities and locations of drivers and rides. However, the matched drivers may not be the global nearest one because of local matching in ORide. Then, Xie et al. [9] utilized the RNE techniques and difference evaluation scheme based on the property-preserving hash to achieve privacy-preserving ride-matching. But, Vivek [10] demonstrated that the scheme faces a passive attack, which can recover locations of riders and matched drivers. Therefore, how to achieve secure ride-hailing becomes a challenge.

To resolve the above problem, a privacy-preserving cross-zone ride-matching scheme for ORH (CRide) is proposed. Based on the coordinate ciphertext of riders and drivers, CRide calculates the distance ciphertext of the rider from all drivers without learning their locations and then sends it to the rider. The distance ciphertext is decrypted and the nearest driver is found by the rider. We conduct an experiment on real dataset to evaluate CRide, and results show that it achieves a high accuracy and acceptable computing and communication efficiency without the risk of privacy disclosure. The primary contributions of this thesis are introduced as follows:

(1) We propose a privacy-preserving cross-zone ride-matching scheme for ORH service (CRide), which can find the nearest driver for a rider. During matching, the location information of drivers and riders is protected. Using the ciphertext packing technology, the proposed scheme simultaneously calculates the distance ciphertext between the rider and all drivers in a single ciphertext, efficiently reducing the bandwidth overhead between the rider and the server.

(2) A ride-matching approach crossing adjacent zones is presented in CRide. It can securely find the nearest driver for riders in the whole zone rather than in the local zone. Thus, the accuracy of ride-matching can be increased.

(3) In order to evaluate the performance and accuracy of CRide, we design and implement the scheme by C++ on real dataset. Experimental results confirm that CRide achieves comparable accuracy with acceptable computation and network cost for ORH.

This paper is organized as follows. The problem statement is given in Section 2. The necessary preliminaries are introduced in Section 3. The privacy-preserving ride-hailing scheme crossing zones is proposed in Section 4. The security of scheme is analyzed in Section 5. CRide is implemented and evaluated in terms of performance and accuracy in Section 6. At last, related works and conclusions are, respectively, given.

2. Problem Statements

2.1. System Model. In ORH service, when a rider requests a ride query, the nearest driver is matched. In the period of matching, riders' and drivers' sensitive information like location and identification is protected. Figure 1 shows the system model which includes three entities, i.e., *RS*, drivers, and riders.

(1) *Riders*. Riders send ride request to *RS*, and ride request includes rider's public key *pk*, coordinate ciphertexts, and located zone. After receiving the distance ciphertexts from *RS*, riders decrypt the ciphertexts and find the nearest driver.

(2) *RS*. *RS* computes the distance ciphertext based on the encrypted coordinates of riders and drivers. Meanwhile, *RS* sends *pk* to drivers who are in the rider's zone z_i and the adjacent zones $z_{i,j}$, where $1 \leq j \leq 8$.

(3) *Drivers*. Drivers register on the *RS* and update their zones to the *RS* when their zones are changed. Drivers in the rider's zone and some adjacent zones send their encrypted location information to *RS* when they receive pk from *RS*.

2.2. Threat Model

(1) *RS* is honest but curious, which is curious about drivers' and riders' sensitive information although following the protocol. So, except for the ride-matching results, locations and identification of riders and drivers cannot be leaked to *RS*.

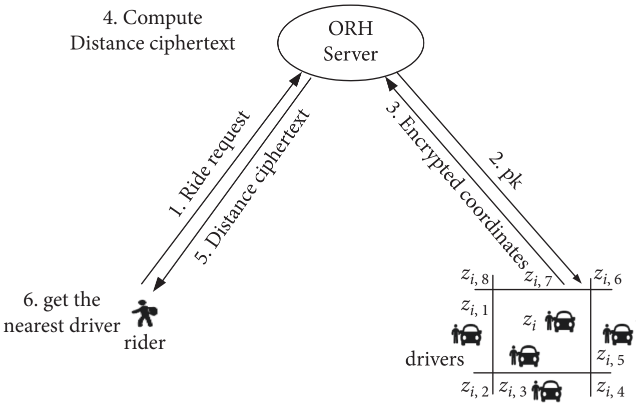

Figure 1: The system model of CRide.

(2) Riders and drivers are also honest but curious. Riders upload ride request to get the nearest driver, and drivers send their encrypted locations to *RS*, but they are curious about the location information of each other. Meanwhile, drivers may be ill-natured who could destroy other drivers' location information.

(3) There may be adversaries from Internet, which may launch eavesdropping to learn sensitive locations. Therefore, the information of drivers and riders ought to be protected, about which nontrusted entity should not learn anything.

(4) Drivers do not collude with *RS*. The assumption is reasonable because drivers are not the employees of ORH service.

2.3. Design Goals. Based on the above scenarios and the threat model analysis, the scheme will perform ride-hailing matching efficiently while protecting the privacy of drivers and riders. The specific goals are described here.

(1) *Security*. During the matching process, CRide ought to protect the privacy of drivers and riders. *RS* only knows the riding-matching result and does not learn anything else. Meanwhile, riders and drivers also learn nothing about the locations of each other except the matched driver.

(2) *Accuracy*. The scheme should be able to match the nearest driver accurately according to the distance between riders and drivers in the whole zone.

(3) *Efficiency*. CRide should achieve a high ride-matching accuracy rate with acceptable costs of communication and computation in practice.

3. Preliminaries

3.1. Somewhat Homomorphic Encryption. Homomorphic encryption allows people to perform any operations on the encrypted data and get encrypted results whose plaintext is the same as the operations conducted in plain domain

[11, 12]. Throughout the whole operation process, ciphertext need not be decrypted to protect data privacy. It is crucial for *RS* to compute the match value based on encrypted location coordinates of drivers and riders while not leaking plaintext. In our scheme, we use the somewhat homomorphic encryption (SHE) which supports some types of operations on ciphertexts with limited times.

As an efficient SHE, the Fan–Vercauteren (*FV*) scheme [13] is an additively and multiplicatively homomorphic encryption. In particular, plaintext m and ciphertext c are polynomials over a ring, plaintext elements $m \epsilon R_t = Z_t[X]/(X^d + 1)$, ciphertext elements $c \epsilon R_q = Z_q[X]/(X^d + 1)$, t and q are positive integers which, respectively, define the maximum of the plaintext and ciphertext coefficients, and $q > t$.

(1) FV.KeyGen (1^λ): suppose $_\chi$ is a short noise random distribution in R_q, $s \leftarrow \chi$ is a secret key sk, a is a random element in R_q, and $e \leftarrow \chi$ is a noise term; then, output pubic key $pk = [p0, p1] = ([-(a.s + e)]_q, a)$.

(2) FV.Enc$_{pk}(m)$: to encrypt a message $m \epsilon R_t$, let $\Delta = q/t$, sample $u, e_1, e_2 \leftarrow \chi$ and return $c = ([p_0.u + e_1 + \Delta \cdot m]_q,\ [p_1.u + e_2]_q) = [c_0, c_1]$.

(3) FV.Dec$_{sk}(c)$: decrypt ciphertext c with secret key sk, $m = (\lceil t.[c_0 + c_1.s \bmod q]/q \rceil \bmod|t)$.

3.2. Number-Theoretic Transform. Number-theoretic transform (*NTT*) is a Fourier transform for finite fields which can speed up the polynomial multiplication in encrypted domain and transform convolution products into coefficient-wise products [14]. For a vector $x \in Z_t^n$, its *NTT* is represented as

$$X = [NTT(x)_k]_{k=0}^{n-1} = \left[\sum_{i=1}^{n-1} x_i \alpha^{ki}\right]_{k=0}^{n-1}. \tag{1}$$

Inverse NTT:

$$x = [NTT^{-1}(X)_k]_{k=0}^{n-1} = \left[n^{-1} \sum_{k=1}^{n-1} X_k \alpha^{-ki}\right]_{i=0}^{n-1}, \tag{2}$$

where n^{-1} is the modulo inverse of n in Z_t and α is a principal nth root of unity in Z_t.

4. Privacy-Preserving Ride-Matching Scheme Crossing Zones

4.1. CRide Overview. Generally, both riders and drivers can use smartphones with third-party navigation apps and map projection systems to get their locations and convert the pair of (latitude, longitude) to planar coordinates (x, y). To lighten the calculation and communication burden, we can divide a city into a number of zones. When a rider needs ride-hailing, he sends his encrypted planar coordinates to *RS*. *RS* packs coordinate ciphertexts into a ciphertext polynomial after receiving coordinate ciphertexts of drivers in the local zone. Then, the distance ciphertext of the rider from all drivers is simultaneously calculated in a single ciphertext.

When the rider receives the distance ciphertext, he decrypts it and gets the minimum distance value *dist*. Based on the *dist*, the rider finds the adjacent zones. For each adjacent zone, the same operation is performed iteratively. Finally, the nearest driver in the whole zone is matched.

Note that it is impossible that a driver currently being matched is matched to a new rider again, so we use variable matched$_{[k]}$ to represent the status of the kth driver. If matched$_{[k]} = 1$, the kth driver is busy; otherwise, he is free. Drivers with busy status will not be considered for a new ride requesting.

4.2. Framework of CRide. Figure 2 depicts the framework of CRide which can be described as follows:

(1) *Initialization.* A city is divided into a number of zones of a certain size vertically and horizontally, and some parameters are set. Drivers registered on ORS service submit their zone locations to *RS* and constantly update when their zones are changed. Meanwhile, matched$_{[k]}$ is set to 0.

(2) When a rider requests ride-hailing, he firstly generates a key pair (pk, sk) and then packs his location (x_R, y_R) into two polynomials $P_{x_R} = \sum_{i=0}^{d-1} x_R X^i$, $P_{y_R} = \sum_{i=0}^{d-1} y_R X^i$. Next, he applies inverse *NTT* on polynomial P_{x_R} and P_{y_R} and encrypts them to C_{x_R} and C_{y_R}. Finally, the rider sends pk, his zone location (e.g., z), C_{x_R}, and C_{y_R} to *RS*.

(3) *RS* sends the upper right corner coordinates (x_{rt}, y_{rt}) and the lower left corner coordinates (x_{lb}, y_{lb}) of zone z to the rider.

(4) *RS* assigns each driver in zone z an index i $(0 \leq i \leq n)$ except for those with matched$_{[k]} = 1$. Then, all the indexes and pk are sent to the corresponding drivers.

(5) For the ith driver, his coordinates (x_{D_i}, y_{D_i}) can be encoded in the ith coefficient $q_{x_D}^i = x_{D_i} X^i$ and $q_{y_D}^i = y_{D_i} X^i$. Then, he applies inverse NTT on them and encrypts them to get the ciphertext of coordinates $c_{x\ D}^i$ and $c_{y\ D}^i$. Finally, the driver sends $c_{x\ D}^i$ and $c_{y\ D}^i$ to *RS*. Similar works are requisite for all the drivers.

(6) *RS* packs ciphertext of locations from n drivers into a single ciphertext $(c_{x\ D} = \sum_{i=0}^{n-1} c_{x\ D}^i$ and $c_{y\ D} = \sum_{i=0}^{n-1} c_{y\ D}^i)$ after receiving them. The base $X = 2^\theta$ and θ is large enough to separate each coordinate value at the bit level. Then, *RS* computes the distances among the rider and n drivers in parallel by $C_{di\ st} = (C_{x_R} - C_{x_D})^2 + (C_{y_R} - C_{y_D})^2$ in encrypted domain and sends $C_{di\ st}$ to the rider.

(7) The distance ciphertext is decrypted and NTT is applied to get the distance polynomial, which consists of distances between rider and each driver. Then, the rider gets the shortest distance d^* in local zone. Taking d^*, (x_{rt}, y_{rt}) and (x_{lb}, y_{lb}) as inputs, rider runs algorithm 1 to find adjacent zones array β

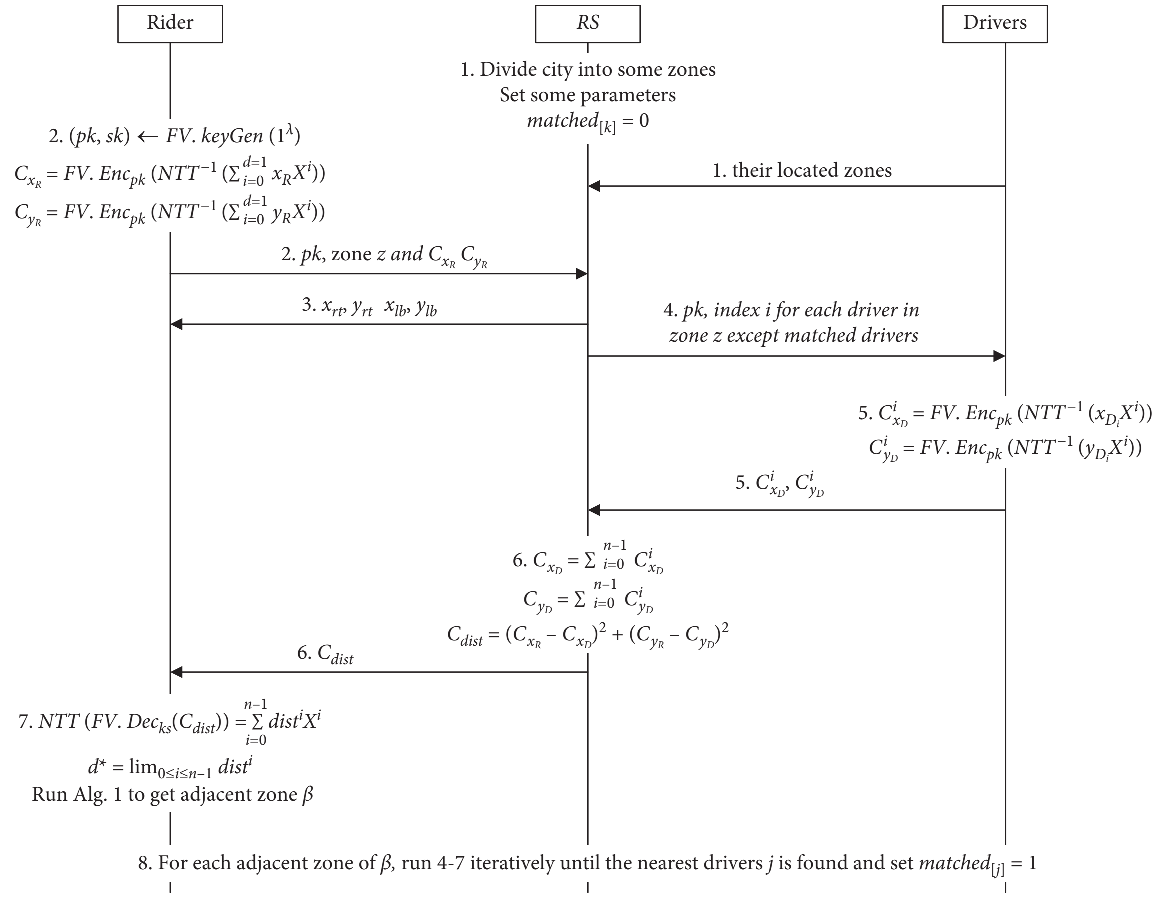
Rider
RS
Drivers
1. Divide city into some zones
Set some parameters
$matched_{[k]} = 0$
2. $(pk, sk) \leftarrow FV.\ keyGen\ (1^{\lambda})$
$C_{x_R} = FV.\ Enc_{pk}\ (NTT^{-1}\ (\Sigma_{i=0}^{d=1}\ x_R X^i))$
$C_{y_R} = FV.\ Enc_{pk}\ (NTT^{-1}\ (\Sigma_{i=0}^{d=1}\ y_R X^i))$
1. their located zones
2. pk, zone z and C_{x_R} C_{y_R}
3. $x_{rt}, y_{rt}\ \ x_{lb}, y_{lb}$
4. pk, $index$ i for $each$ $driver$ in $zone$ z $except$ $matched$ $drivers$
5. $C^i_{x_D} = FV.\ Enc_{pk}\ (NTT^{-1}\ (x_{D_i} X^i))$
$C^i_{y_D} = FV.\ Enc_{pk}\ (NTT^{-1}\ (y_{D_i} X^i))$
5. $C^i_{x_D}, C^i_{y_D}$
6. $C_{x_D} = \Sigma_{i=0}^{n-1}\ C^i_{x_D}$
$C_{y_D} = \Sigma_{i=0}^{n-1}\ C^i_{y_D}$
$C_{dist} = (C_{x_R} - C_{x_D})^2 + (C_{y_R} - C_{y_D})^2$
6. C_{dist}
7. $NTT\ (FV.\ Dec_{ks}(C_{dist})) = \sum_{i=0}^{n-1} dist^i X^i$
$d^{\star} = \lim_{0 \le i \le n-1} dist^i$
Run Alg. 1 to get adjacent zone β
8. For each adjacent zone of β, run 4-7 iteratively until the nearest drivers j is found and set $matched_{[j]} = 1$

Figure 2: The framework of CRide.

with at most 8 zones [15]. Generally, the nearest driver exists in one of them.

(8) For each adjacent zone, 4–7 are run iteratively until the nearest driver is found and its $\text{matched}_{[k]}$ is set to 1.

5. Security Analysis

5.1. Security Analysis from RS. *RS* is honest but curious who follows the protocol but is curious about locations of riders and drivers. Towards privacy-preserving ride-matching, riders and drivers separately encrypt their location coordinates (x_R, y_R) and (x_D, y_D) with public key *pk* and then send individually location ciphertexts (C_{x_R}, C_{y_R}) and (C_{x_D}, C_{y_D}) to *RS*. So, the procedure of ride-matching is conducted in encrypted domain that prevents attackers and *RS* from learning anything because of the semantic security of *FV*. Meanwhile, none of *RS* and attackers has the rider's private key *sk*. So, they cannot decrypt the ciphertext. In the matching process, CRide ought to protect the privacy of drivers and riders. RS only knows the riding-matching result and does not learn anything else. Thus, *RS* cannot launch inference attack on riders.

5.2. Security Analysis from Driver. In the proposed scheme, drivers may be ill-natured who could destroy other drivers' location information by encrypting nonzero values for other slots. To prevent the malicious attack, *RS* sets a different index *i* for each driver, and drivers encode their coordinates in the *i*th coefficient $q^i_{x_D} = x_{D_i} X^i$ and $q^i_{y_D} = y_{D_i} X^i$, by which, *RS* can figure out the malicious behavior.

5.3. Security Analysis from Rider. During ride-matching, drivers' locations are encrypted before being sent to *RS*. *RS* computes the ciphertext of distance and sends it to the rider. After receiving the ciphertext of distance, the rider can decrypt it and find the nearest driver. Meanwhile, he learns nothing about the location of the matched driver except for his *ID,* although the rider knows *sk*.

6. Evaluation

To evaluate communication and computation overload and ride-matching accuracy, some experiments are conducted on real-world dataset. In the experiments, we set the polynomial dimension of a 20-bit plaintext and the coefficient size to 4096 and 124 bits, respectively. Thus, CRide can reach a 112-bit security and support ciphertext calculation of 4096 bits. We implement the scheme with the C++ NFLlib library [16]. RS is deployed on a machine with Intel i7-10700 which has 2.9 GHz CPU and 16 GB RAM. Riders and drivers are located on the same machine with Intel i7-3537U which has 2.5 GHz CPU and 8 GB RAM.

```
Input: the upper right corner coordinates $(x_{rt}, y_{rt})$ and lower left corner $(x_{lb}, y_{lb})$ of zone z
   The distance $d*$ between the nearest driver and the rider in zone $z$
   The coordinate $(x_r, y_r)$ of the rider
Output: zones array $\beta$ that nearest driver may exist
$\beta$ is initialized to empty
if $x_r + \text{dist} > x_{rt}$, then add $z_3$ into $\beta$
if $x_r - di\ st < x_{lb}$, then add $z_7$ into $\beta$
if $y_r - di\ st < y_{lb}$, then add $z_5$ into $\beta$
if $y_r + di\ st > y_{rt}$, then add $z_1$ into $\beta$
if $(x_r - x_{lb})^2 + (y_r - y_{lb})^2 < di\ st^2$, then add $z_6$ into $\beta$
if $(x_r - x_{lb})^2 + (y_r + y_{rt})^2 < di\ st^2$, then add $z_8$ into $\beta$
if $(x_r + x_{rt})^2 + (y_r + y_{rt})^2 < di\ st^2$, then add $z_2$ into $\beta$
if $(x_r + x_{rt})^2 + (y_r - y_{lb})^2 < di\ st^2$, then add $z_4$ into $\beta$
return $\beta$
```

Algorithm 1: Adjacent zone search.

6.1. Dataset. CRide is evaluated based on real dataset that comes from [17]. In the experiment, we divide a zone z from a dataset for October 2014 with over 14 million riders. Zone z has about 4096 drivers and is used to evaluate the performance. Towards cross-zone ride-matching, zone z is divided into some zones vertically and horizontally. For example, the zone is divided into nine zones of the same size using $3 * 3$ divisions.

In the dataset, we assume a record is a ride-matching request, and the pick-up location of a rider is his location. Since 99% of the time interval between the drop-off of a driver and his next pick-up is about 30 seconds, a driver is available for a ride request if there is once drop-off in the last 30 seconds.

6.2. Performance. This part evaluates the performance of CRide, including communication and computation overhead compared with PrivateRide [7] and ORide [8]. Based on the above settings, the polynomial size is 62 KB. The maximum number of available drivers for a request is 4096. Experimental results are collected after 100 trials. When a rider requests ride-matching, communication overhead and computation cost are, respectively, shown in Tables 1 and 2.

6.2.1. The Performance of Riders. Assuming that there are n drivers and m adjacent zones, the nearest driver may exists in one of zones. The rider sends *RS* a public key and two encrypted coordinates with six polynomials and a 372 KB payload for a ride request. The download overhead is linear to the number of available drivers in PrivateRide. With the ciphertext packing technique, ORide only has a distance ciphertext, and the number of distance ciphertext is reduced to $m * n/d]$ in CRide. The size of an encrypted distance is 186 KB. For a ride request, 4096 encrypted distances are sent to the rider in RrivateRide. In CRide, the download overhead is linear to the number of chosen adjacent zones. The maximum value of m is 8. Experiments show that m is up to 3 in most cases. One hundred experiments show that on average, *RS* sends the rider the ciphertext of distance with size 353.4 KB for 3 * 3 division. Table 1 shows that the communication overhead of CRide is slightly increasing compared with ORide, which is significantly reducing compared with PrivateRide.

Table 1: Per-ride communication overhead evaluation.

Scheme	Rider Comm (KB)	Driver Comm (KB)	RSUs Comm (MB)
PrivateRide	762,228.0	372	1,736.5
ORide	558.0	372	992.7
CRide (3 ∗ 3)	725.4	372	353.9

As shown in Table 2, the three schemes have the same computation cost for key generation and encryption. The decryption cost of CRide increases slightly compared with ORide. However, it is also significantly reduced compared with PrivateRide.

6.2.2. The Performance of Drivers. Tables 1 and 2 show that the three schemes have the same communication and computational overhead for each ride request. For the driver, he will receive the public key pk and upload his coordinate encryption for ride requests from his zone and some adjacent zones. The driver's bandwidth should meet the ride-matching while not being too large. The bandwidth is calculated by multiplying the size of the required bandwidth per request with the number of receiving requests per second. Results of the experiment show that the required bandwidths for CRide (3 * 3), CRide (6* 6), and CRide (9* 9) are separately less than 0.68 Mb/s, 0.3 Mb/s, and 0.23 Mb/s. Therefore, RS should balance the bandwidth requirement for drivers and zone division.

6.2.3. The Performance of RS. As shown in Table 1, using ciphertext packing and cross-zone ride-matching, the communication overhead of CRide is significantly reduced compared with PrivateRide and ORide.

The computation cost of CRide is linear to the number m of adjacent zones, and that of PrivateRide is linear to the number of available drivers. Table 2 shows that the computation cost of CRide is significantly reduced compared to

Table 2: Per-ride computation cost evaluation.

	Rider			Driver		RSUs	
Scheme	Comp (s)			Comp (s)		Comp (s)	
	Genkey (ms)	Enc (ms)	Dec (ms)	Loadkey (ms)	Enc (ms)	Loadkey (ms)	CompDist (ms)
PrivateRide	1.34	2.39	6,212.44	0.36	2.39	0.29	98,665.60
ORide	1.34	2.39	1.89	0.36	2.39	0.29	187.81
CRide (3 ∗ 3)	1.34	2.39	3.591	0.36	2.39	0.29	356.82

PrivateRide, and it is slightly increasing compared with ORide.

6.3. Accuracy. For ride-hailing service, the key is to match the nearest drivers for riders. In experiments, 100 ride requests are generated randomly and implemented. We use ride-matching accuracy rate to demonstrate the matching degree of PrivateRide, ORide, and CRide with the ground in plaintext. As shown in Figure 3, CRide gets the nearest drivers for about 92% of riders, which exceeds ORide by nearly 30% under all division granularities, which implements ride-matching without cross zone. Meanwhile, CRide reaches the same accuracy as PrivateRide, which implements ride-matching without division because of cross-zone matching. But Table 1 shows that the communication overhead and computation cost of PrivateRide are considerably large compared with CRide.

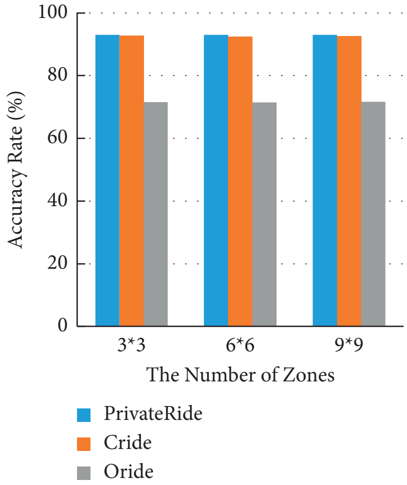

Figure 3: The matching accuracy with different division granularities.

7. Related Work

With the popularity of ride-hailing services, privacy protection receives more and more attention. Some works have been done to design privacy-preserving ride-hailing schemes. Duan et al. and Khazbak et al. [18, 19] used cloaking technologies to match the nearest drivers for a ride request while protecting the sensitive information of drivers and riders. Duan et al. [18] proposed cloaking region-based passenger privacy protection in ride-hailing, maximizing social welfare under riders' privacy requirements. Khazbak et al. [19] proposed a ride-hailing scheme with privacy preserving, which considers the privacy preference of riders using novel obfuscation techniques. However, these schemes use location range to find the nearest drivers for a rider, which results in low accuracy.

To promote the accuracy of ride-matching, schemes [15, 20] were designed using the road network embedding technology (RNE) to match. Luo et al. [15] proposed pRide which allows ORH to efficiently match riders and drivers based on RNE and garbled circuits. Yu et al. [20] proposed RMatch, which computes the shortest distance using RNE and PHE. Meanwhile, Yu et al. [20] also proposed EPRide, in which the Hamming distance replaces road distance to implement ride-matching using somewhat homomorphic encryption and load network hypercube embedding technology. But the three schemes assume that there is a trusted party, which cannot fully guarantee the privacy of user in practice.

There are new works on privacy-preserving ride-hailing. Shivers et al. [21] proposed a framework for developing a decentralized ride-hailing architecture for autonomous vehicles implemented on the Hyperledger Fabric blockchain platform. Huang et al. [22] proposed a privacy-preserving ride-matching scheme with prediction, which utilizes a deep learning model to predict the emergence of ride requests in various regions and find the best driver in a global perspective instead of the nearest driver in the local region, leveraging prediction results.

8. Conclusions

In this thesis, a privacy-preserving cross-zone ride-matching scheme is proposed in ride-hailing services. The city is divided into some zones, and ride-matching is implemented in cross adjacent zones using somewhat homomorphic encryption and ciphertext packing. By using CRide, *RS* can find the nearest driver for riders in the whole zone. During matching, the privacy of drivers and riders is protected. Theoretical analysis and experimental results over real-world datasets prove that CRide achieves acceptable efficiency and comparable matching accuracy with the ground in plaintext. In the next research, the reputation of drivers and the interest of riders are worthy of considering for ORH service.

Data Availability

The data that support the findings of this study are from previously reported studies and datasets, which have been cited. The processed data are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This study was supported by the Key Technologies R&D Program of He'nan Province under grant nos. 192102210295 and 212102210084.

References

[1] Uber technologies inc, "Uber technologies inc," 2021, https://www.uber.com/.

[2] Didi, "Didiglobal," 2020, https://www.didiglobal.com/.

[3] S. Vivek, "Attacks on a privacy-preserving publish-subscribe system and a ride-hailing serving," in *Proceedings of the 18th IMA International Confirerence on Cryptogrphy and Coding, Virtual*, pp. 1–15, Manhattan, NY, USA, December 2021.

[4] D. Kumaraswamy and S. Vivek, "Cryptanalysis of the privacy-preserving ride-hailing service TRACE," in *Proceedings of the 22th International Conference on Cryptology*, pp. 462–484, Jaipur, India, December 2021.

[5] J. Ni, K. Zhang, X. Lin, and X. Shen, "Securing fog computing for Internet of things applications: challenges and solutions," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 1, pp. 601–628, 2018.

[6] H. Yu, H. Zhang, X. Yu, X. Du, and M. Guizani, "PGRide: privacy-preserving group ridesharing matching in online ride hailing services," *IEEE Internet of Things Journal*, vol. 8, no. 7, pp. 5722–5735, 2020.

[7] A. Pham, I. Dacosta, B. Jacot-Guillarmod, and H. Kévin, "PrivateRide: a privacy-enhanced ride-hailing service," in *Proceedings of the Privacy Enhancing Technologies Symposium*, pp. 38–56, Minneapolis, USA, April 2017.

[8] A. Pham, I. Dacosta, G. Endignoux, J. Troncoso-Pastoriza, K. Huguenin, and J. Hubaux, "ORide: a privacy-preserving yet accountable ride-hailing service," in *Proceedings of the 26th USENIX Security Symposium*, pp. 1235–1252, Vancouver, BC, Canada, August 2017.

[9] H. Xie, Y. Guo, and X. Jia, "A privacy-preserving online ride-hailing system without involving a third trusted server," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 3068–3081, 2021.

[10] S. Vivek, "Comments on "A Privacy-Preserving Online Ride-Hailing System without Involving a Third Trusted Server," 2021, https://arxiv.org/abs/2112.06449.

[11] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *Proceedings of the EUROCRYPT'99 on Theory and Application of Cryptographic Techniques*, pp. 223–238, Prague, Czech Republic, 2-6 May 1999.

[12] Z. Li, X. Gui, and Y. Gu, "Survey on homomorphic encryption algorithm and its application in the privacy-preserving for cloud computing," *Journal of Software*, vol. 29, no. 7, pp. 1830–1851, 2018.

[13] J. Fan and F. Vercauteren, "Somewhat Practical Fully Homomorphic Encryption," *Iacr Cryptology EPrint Archive*, vol. 2012, p. 144, 2012.

[14] A. Pedrouzo-Ulloa, J. R. Troncoso-Pastoriza, and F. Perez-Gonzalez, "Number theoretic transforms for secure signal processing," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 5, pp. 1125–1140, 2017.

[15] Y. Luo, S. Wang, K. Ren et al., "Transition-metal dichalcogenides/Mg(OH)2 van der Waals heterostructures as promising water-splitting photocatalysts: a first-principles study," *Physical Chemistry Chemical Physics : Physical Chemistry Chemical Physics*, vol. 21, no. 4, pp. 1791–1796, 2019.

[16] NFLlib, "NFLlib," 2016, https://github.com/quarkslab/NFLlib.

[17] Dataset, "Dataset," 2022, https://github.com/toddwschneider/nyc-taxi-data.

[18] Y. Duan, G.-J. Gao, M.-J. Xiao, and J. Wu, "Cloaking region based passenger privacy protection in ride-hailing systems," *Journal of Computer Science and Technology*, vol. 35, no. 3, pp. 629–646, 2020.

[19] Y. Khazbak, J. Fan, S. Zhu, and G. Cao, "Preserving personalized location privacy in ride-hailing service," *Tsinghua Science and Technology*, vol. 25, no. 6, pp. 743–757, 2020.

[20] H. Yu, J.-Q. Li, L. Zhang, and P. Duan, "An imperialist competition algorithm using a global search strategy for physical examination scheduling," *Applied Intelligence*, vol. 51, no. 6, pp. 3936–3951, 2020.

[21] R. Shivers, M. A. Rahman, M. J. Hossain, H. Shahriar, and A. Cu, "Ride-hailing for autonomous Vehicles: hyperledger fabric-based secure and decentralize Blockchain platform," in *Proceedings of the 2021 IEEE International Conference on Big Data*, pp. 1–11, Orlando, FL, USA, December 2021.

[22] J. Huang, Y. Luo, S. Fu, M. Xu, and B. Hu, "pRide: privacy-preserving online ride hailing matching system with prediction," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 8, pp. 7413–7425, 2021.