





## Article

# Decoupling Online Ride-Hailing Services: A Privacy Protection Scheme Based on Decentralized Identity

Nigang Sun <sup>1</sup>, Yuxuan Liu <sup>2,\*</sup>, Yuanyi Zhang <sup>3</sup> and Yining Liu <sup>4</sup>

<sup>1</sup> School of Microelectronics and Control Engineering, Changzhou University, Changzhou 213000, China; ngsun@cczu.edu.cn

<sup>2</sup> School of Computer Science and Artificial Intelligence, Changzhou University, Changzhou 213000, China

<sup>3</sup> Shanghai Shentie Information Engineering Co., Ltd., No.12, Huanchengdong Road, Shangcheng District, Hangzhou 310009, China; revanton@icloud.com

<sup>4</sup> School of Computer and Information Security, Guilin University of Electronic Technology, Guilin 541004, China; ynliu@guet.edu.cn

\* Correspondence: s23150812020@smail.cczu.edu.cn

**Abstract:** Online ride-hailing services have become a vital component of urban transportation worldwide due to their convenience and flexibility. However, the expansion of their user base has dramatically heightened the risks of user privacy information leakage. Among these risks, the privacy leakage problem caused by the direct correlation between user (driver and passenger) identity information and location-based ride information is of particular concern. This paper proposes a novel privacy protection scheme for ride-hailing services. In this scheme, decentralized identities are employed for user authentication, separating the identity registration service from the ride-hailing platform, thereby preventing the platform from obtaining user privacy data. The scheme also employs a fuzzy matching strategy based on location Points of Interest (POI) and a ciphertext-policy attribute-based hybrid encryption algorithm to hide the user's precise location and restrict access to location information. Crucially, the scheme achieves the complete decoupling of identity registration services and location-based ride services on the ride-hailing platform, ensuring that users' real identities and ride data cannot be directly associated, effectively protecting user privacy. Within the decoupled architecture, regulatory authorities are established to handle emergencies within ride-hailing services. Through simulation experiments and security analysis, this scheme is demonstrated to be both feasible and practical, providing a new privacy protection solution for the ride-hailing industry.

**Keywords:** online ride-hailing; privacy protection; blockchain; decentralized identity; fuzzy location matching; ciphertext-policy attribute-based hybrid encryption



**Citation:** Sun, N.; Liu, Y.; Zhang, Y.; Liu, Y. Decoupling Online Ride-Hailing Services: A Privacy Protection Scheme Based on Decentralized Identity. *Electronics* **2024**, *13*, 4060. <https://doi.org/10.3390/electronics13204060>

Academic Editors: Mehdi Sookhak and Fabio Grandi

Received: 18 September 2024

Revised: 8 October 2024

Accepted: 14 October 2024

Published: 15 October 2024



**Copyright:** © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Online ride-hailing services have become a part of people's daily lives due to their convenient reservation, reasonable prices, and technical support, meeting the needs of fast, safe, and flexible travel in modern urban life [1]. However, with the growing user base, the privacy and security issues caused by the collection and sharing of user data, such as travel history, precise location, and personal information, exposed by ride-hailing services during their usage, have become increasingly prominent [2]. For instance, in October 2016, Uber faced two major security incidents. First, the company was accused of both collecting a vast amount of user data, including names, usernames, emails, access devices, etc., and of allowing all employees unfettered access to ride data, leading to severe privacy violations [3]. Almost simultaneously, a severe hacker attack led to the leakage of personal information, such as names, emails, phone numbers, etc., of 57 million Uber users worldwide [4]. These serious privacy issues have spurred numerous solutions to protect user privacy, which can be categorized into three types: obfuscating key information, decentralized storage,

and digital identity. Obfuscating key information is a data desensitization technique that reduces the risk of privacy leakage by hiding, modifying, or generalizing detailed data. When applied to protect location information, this technique typically employs various methods, such as hiding specific locations within larger geographical areas, adding random noise, reducing precision, or using pseudonymization techniques. These methods effectively protect location privacy while maintaining a certain degree of data usability. Chow et al. [5] proposed a spatial cloaking technique based on a Trusted Third Party (TTP) to obfuscate user location information. Hengartner [6] proposed obfuscating user location by combining Private Information Retrieval (PIR) and trusted computing. However, obfuscating key information only safeguards location privacy, while service providers possess users' true identities, potentially leading to severe privacy breaches. Decentralized storage leverages distributed computing technology to enhance data security. As demonstrated in [7], the traditional ride-hailing system characterized by centralized data storage inherently faces vulnerabilities to data manipulation and single-point intrusion due to its storage of various pieces of sensitive information and personal data of users. In contrast, decentralized storage mitigates the risks associated with data centralization in the ride-hailing system. Renu et al. [8] proposed a privacy protection scheme for ride-hailing services based on decentralized storage and smart contracts. Similarly, Fadhil et al. [9] introduced a solution that safeguards passenger privacy through blockchain smart contracts and spatial cloaking techniques. Nonetheless, such decentralized storage still encounters the challenge of service providers excessively accumulating user identity information; thereby, users still face the risk of privacy leakage. Digital identity represents a mapping of individuals' real identities in digital space. With the rise of blockchain technology, this mapping process has been endowed with unprecedented security, transparency, and decentralization. In ride-hailing services, digital identity can be used for secure user verification and privacy-preserving transactions. The schemes [10–12] have formulated innovative digital identity solutions, harnessing the unique strengths of blockchain technology, which not only safeguard the security of user identity privacy but also offer reliable identity verification. The utilization of digital identity allows for precise control over data, effectively averting the leakage of superfluous details unrelated to the required identity verification, thus compensating for the privacy leakage risks caused by the lack of identity information protection in the previous two types of solution. Consequently, digital identities have garnered widespread adoption in the realm of ride-hailing privacy protection.

Recent studies [13–16] indicate that the use of digital identities has spawned a variety of privacy protection schemes. Sánchez et al. [17] proposed a decentralized Peer-to-Peer (P2P) ride-sharing system, utilizing anonymous node ID to conceal users' true identities. However, this system requires drivers and passengers to reveal their real identities to each other. Kang et al. [18] introduced a privacy-preserving scheme based on pseudonyms, which protected vehicle owners' privacy when broadcasting information by employing pseudonym changes and secure pseudonym management protocols. Pham et al. [19] proposed a ride-hailing service system called PrivateRide, which used blind signatures and anonymous credentials as digital identities to safeguard passenger privacy. Hong et al. [20] proposed OCHJRNChain, a blockchain-based secure data-sharing framework, which employed zero-knowledge proof technology to allow passengers to prove their identity effectively and securely without revealing any additional personal information. This framework also incorporated blockchain and homomorphic encryption technologies to create a system that protected passenger privacy. However, regrettably, the schemes proposed by Pham et al. and Hong et al. do not provide drivers with the same level of privacy protection as passengers and fail to address the challenges posed by drivers' information to passenger privacy. Shen et al. [21] introduced a ride-hailing privacy protection scheme without the need for a trusted third party. This scheme used public keys as user identity identifiers and combined Public Key Encryption with Equality Test (PKEET) and blockchain smart contract technologies to achieve ride-hailing matching. Although this highly encrypted method enhanced user privacy protection, it made it difficult to trace

the identities and vague location information of drivers and passengers in emergency situations. Pham et al. [22] proposed a ride-hailing system based on homomorphic encryption technology, which used digital certificates and anonymous credentials to protect user identity information, allowing service providers to match drivers and passengers without directly accessing their private information. However, service providers can map credentials to identities and track users through random IDs, posing a certain degree of threat to user privacy. Decentralized identities have been widely applied in privacy protection strategies due to their ability to grant individuals autonomous control over their data management. Maram et al. [23] proposed CANDID, a decentralized authentication platform independent of TTP. Kang et al. [24] presented an identity management system that combined the Cheon–Kim–Kim–Song (CKKS) fully homomorphic encryption (FHE) scheme to ensure user privacy protection. Javed et al. [25] introduced a blockchain-based decentralized identity management system called Health-ID, aiming to securely identify and authenticate users' identities. Stockburger et al. [26] proposed a decentralized identity management system based on Self-Sovereign Identity (SSI) for the public transportation sector to protect passenger privacy. It is noteworthy that the existing online ride-hailing privacy protection scheme faces a primary issue: the direct association of digital identities with users' real identities within the ride-hailing platforms. This direct linkage results in key service information and user privacy data remaining connected within the platforms, thereby leaving users still vulnerable to the risk of privacy breaches.

This paper proposes a novel privacy-preserving scheme for ride-hailing services, utilizing decentralized identities, fuzzy location matching technology, and ciphertext-policy attribute-based hybrid encryption algorithm to solve the privacy leakage problem caused by the association between service information and privacy data. The scheme improves upon decentralized identities by reconstructing the issuance process of Decentralized Identifiers (DIDs) and Verifiable Credentials (VCs) within the decentralized identity system. In the initial phase of this scheme, government authorities provide these credentials to users after stringent identity verification, which are then used for authentication on the ride-hailing platform, both separating the identity registration service from the platform and enhancing the credibility of identity information. The matching technology effectively leverages POI data from drivers and passengers to identify drivers who are near the passenger, without requiring precise location information. Additionally, during the process of encrypting data, the scheme sets access policies based on driver attributes to restrict access to location data, thereby safeguarding privacy. The scheme ensures that users' authentic identity information is not associated with the ride information within the platform, and it establishes additional regulatory authorities to address emergent hazards in the ride-hailing service, which achieves a complete decoupling of the ride-hailing's key services while reinforcing the regulatory mechanism, effectively protecting users' privacy data.

The remainder of this paper is organized as follows: Section 2 introduces related preparatory work on privacy protection. Section 3 describes the system model and design goals. Section 4 presents the creation process of decentralized identities and their application to ride-hailing platform authentication. Section 5 details the ride-matching scheme of the ride-hailing platform. Section 6 provides a performance analysis of the proposed scheme. Section 7 discusses and analyzes key issues related to user privacy, enterprise risk management, and regulatory considerations. Finally, Section 8 concludes the paper.

## 2. Preliminaries

### 2.1. Blockchain & Smart Contracts

The concept of blockchain was first introduced in "Bitcoin: A Peer-to-Peer Electronic Cash System" [27]. As a type of distributed database technology, blockchain links data blocks in chronological order. Each block contains a batch of data connected through cryptographic technology, forming a continuous and irreversible chain. The core characteristics of blockchain include decentralization, immutability, transparency, security,

and smart contracts [28]. Smart contracts are executable codes that run on the blockchain and automatically execute when predetermined conditions are met, without the need for intermediaries or third-party institutions [29]. This automation enhances the efficiency and security of transactions. The introduction of smart contracts not only enhances blockchain's capabilities, enabling it to automatically execute complex logic, but also expands its application areas, such as supply chain management, financial services, and other commercial sectors [30,31]. By deploying smart contracts, blockchain is no longer just a data storage technology; it has become a broad transaction processing platform, creating a trading environment without the need for trusted third parties, significantly enhancing the application value and potential of blockchain technology. Moreover, blockchain technology, with its transparent, irreversible, and decentralized characteristics, provides a vast development prospect for smart contracts.

### 2.2. Decentralized Identity

Decentralized identity was proposed by the W3C as an identity protocol intended to identify any entity. It mainly consists of decentralized identifiers, DID documents, and verifiable credentials [32]. Each decentralized Identifier is a unique identifier, cryptographically linked to a DID document. The DID documents contain public keys and service endpoints. Verifiable credentials serve as certifications of identity, qualifications, etc., and their authenticity and validity can be verified by any third party through a decentralized network. W3C provides corresponding technical standards and specifications to ensure the system's interoperability and security [33–35].

### 2.3. MinHash & LSH

In 1998, Broder et al. proposed MinHash as a technique for approximating set similarity calculations [36]. The basic idea was to hash the elements of a set into a smaller signature to quickly compare the similarity between two sets. Today, it is widely used for rapid similarity estimation in large datasets, especially in fields like image and text processing. Local Sensitive Hashing (LSH) [37] is an approximate search technique used for quickly finding similar items in high-dimensional spaces, suitable for handling large-scale datasets. The core concept of the LSH algorithm is to map similar elements to the same bucket, thereby speeding up the search process among these elements. By integrating both techniques, the algorithm maps collections to MinHash signatures and utilizes LSH functions to hash similar signatures into the same buckets, enabling efficient retrieval of similar sets or elements within large-scale datasets.

### 2.4. Hybrid Encryption Algorithm

Ciphertext-Policy Attribute-Based Encryption (CP-ABE), proposed by John Bethencourt et al. [38], is a modification and expansion of the earlier proposed Attribute-Based Encryption (ABE). In CP-ABE, attributes describe specific user characteristics and authorities generate private keys for users based on these attributes. Users can only decrypt ciphertexts that match their attribute set, and encryptors set access policies based on specific user attributes as leaf nodes in an access tree. These policies often use logical expressions with thresholds to define the required combination and number of attributes needed for decryption. To enhance security, AES [39] can be integrated to form a hybrid encryption method. Encrypting plaintext with AES and then encrypting the parameter of a symmetric key with CP-ABE provides enhanced data protection.

### 2.5. Spatial Cloaking

Spatial cloaking is a technique designed to protect user location privacy while providing Location-Based Services (LBS). This technique obscures the user's precise location by placing their actual location information within a larger geographic area, thereby protecting their privacy. Common methods employed include spatiotemporal blurring [40] and the k-anonymity model [41]. Spatiotemporal blurring protects user privacy by blur-

ring data in both geographic and temporal dimensions. This approach not only enhances user anonymity when utilizing location-based services but also increases privacy security during data transmission. The  $k$ -anonymity model is a widely used data anonymization technique for privacy protection. Its core principle is to blend the user's location data with that of at least  $k-1$  other users, making it impossible to identify any single user's location information. By integrating spatiotemporal cloaking and the  $k$ -anonymity model, spatial cloaking technology effectively safeguards user privacy while delivering high-quality location-based services.

### 3. Model and Design Goals

This section will analyze from two perspectives: system model and design goals.

#### 3.1. System Model

As illustrated in Figure 1, the model comprises five entities: government departments, passengers, drivers, online ride-hailing platforms, and regulators. Drivers and passengers obtain decentralized identities from the government departments and complete their authentication with the ride-hailing platforms, after which they can begin to offer or request ride-hailing services. Regulatory agencies are responsible for responding to emergencies that arise during the service process. The specific details are as follows.

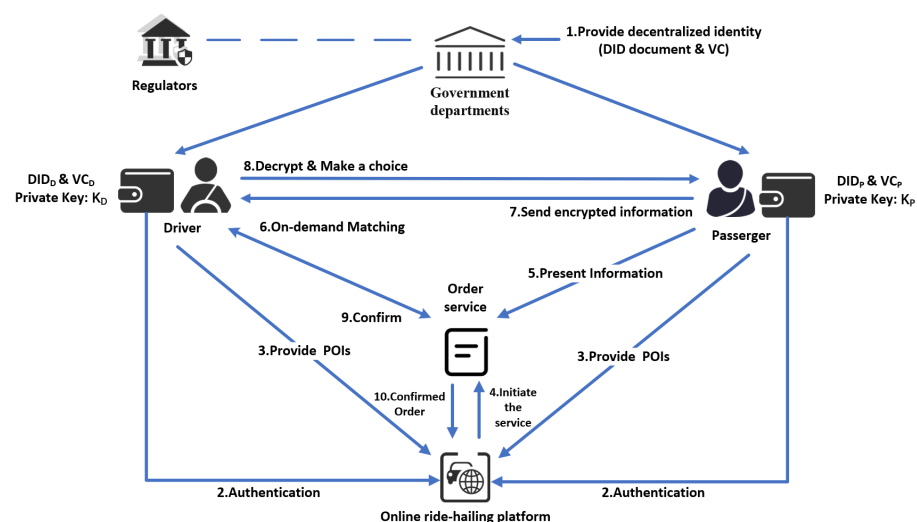


Figure 1. System architecture.

- Government Departments: Government departments receive and verify users' identity information, issuing decentralized identities to users. Additionally, they provide digital wallets for managing the aforementioned information.
- Passengers: Passengers submit their current POI and requirement information, offering encrypted location details to drivers who meet their needs.
- Drivers: Drivers submit their current POI information, decrypt passengers' locations if their attributes match needs, and accept orders.
- Online Ride-hailing Platforms: The ride-hailing platforms aggregate POI information from both passengers and drivers, initiate order matching services to find nearby drivers who meet the requirements, and generate final orders.
- Regulators: As the government entity responsible for criminal investigations, regulators collaborate with government departments and ride-hailing platforms to track drivers and passengers during emergencies by accessing relevant order information. They establish and enforce strict data protection policies to prevent misuse and unauthorized access to sensitive information. Regulators help balance the need for safety and privacy, ensuring that data are accessed only when necessary and for legitimate purposes.

### 3.2. Design Goals

According to the system model proposed above, the scheme proposes an online ride-hailing system based on decentralized identity to decouple key ride-hailing services. The specific design goals are as follows.

1. Identity Autonomy: Users' identity information is in their own hands, and they can provide the information required for verification by the online ride-hailing platform based on the principle of data minimization.
2. Secure Matching: The platform provides matching services to users without accessing their exact locations, thereby fully protecting their location privacy.
3. Uniqueness: Only the driver who meets specific attribute requirements and confirms the order based on the fuzzy location information initially obtained after decryption can obtain the passenger's accurate location information.
4. Service Decoupling: The proposed scheme should separate key services within the online ride-hailing platform that may lead to the association of user data from the platform, protecting user privacy while limiting the power of the online ride-hailing platform.

## 4. Registration

In this section, the process of generating and utilizing decentralized identities is elaborated upon, drawing on the immutability of blockchain to enable the reliable verification of identity information while ensuring utmost protection of user privacy.

### 4.1. Identity Acquisition

#### 4.1.1. Key Pair Generation

Users run a key generation tool provided by government agencies on their local devices. This tool generates a key pair based on the ECC [42] scheme with the following parameters. It first employs the secure cryptography library Bouncy Castle to construct the SECP256K1 elliptic curve  $E_p$ , whose equation is  $y^2 = (x^3 + a \cdot x + b) \bmod p$ , where  $a$  and  $b$  are the curve coefficients and  $p$  is a large prime number defining the field over which the curve is defined. Then, it identifies the base point  $G$  of the curve and a prime number  $q$  as the order of the group generated by  $G$ . A private key  $d$  is generated, satisfying  $0 < d < q$ , and the public key  $P_A$  is computed through point multiplication on the elliptic curve. The parameters  $(p, a, b, q, G)$  constitute the global parameters.  $d$  is stored in the user's digital wallet and is not accessible externally.  $P_A$  is used for subsequent registration and identity verification, which is shown in Equation (1).

$$P_A = d \times G \quad (1)$$

#### 4.1.2. Generation of DID

Before submitting the registration information, the local device will generate a DID for the user. First, the SHA256 hash function is used to hash the decoded public key, producing a 256-bit (32-byte) hash value. This hash is then further processed by the RIPEMD-160 hash function, producing a 160-bit (20-byte) value that shortens the hash length while maintaining its uniqueness and security. Finally, the hash value is encoded using Base58, a coding method commonly used for Bitcoin and other cryptocurrency wallet addresses, resulting in the generation of a corresponding DID for the user, as shown in Equation (2). The scheme assumes that the government has introduced a method called Govchain, which is used for simulation experiments.

$$DID = did : Govchain : Base58(RIPEMD160(SHA256(P_A))) \quad (2)$$

#### 4.1.3. Identity Registration

During the registration process, users need to submit necessary identity information, including DID and public key, and to specify whether they are registering as a 'driver' or a 'passenger'. If registering as a driver, additional information such as a driver's license

and vehicle details must be provided. All of this necessary information will be verified by government agencies.

#### 4.1.4. Generation of DID Document and VC

Government departments upload the DID documents, which are generated using the Govchain method shown in Table 1, to the InterPlanetary File System (IPFS) and then pin them using a Pinning service. IPFS is a protocol based on content addressing and a peer-to-peer network, and files are not directly stored on specific nodes. Depending on the size of the file, it may be split into multiple chunks, each generating a unique Content Identifier (CID). These CIDs can be used to retrieve the file contents through the IPFS network. To ensure the long-term availability of the file, after uploading the data the server pins the file’s CID through a Pinning service, ensuring the corresponding file chunks are retained on persistent storage nodes. Even if the original upload node goes offline, users can still access the file content through the IPFS gateway using the generated CID. The corresponding document index CID and timestamp are stored in the blockchain through Algorithm 1. To meet the identity verification requirements of ride-hailing platforms, government departments and the platforms negotiate the scheme for verifiable credentials. Based on user-submitted data, government departments generate VCs that only contain non-privacy-sensitive information, ensuring they meet the platform’s business needs while avoiding the acquisition of users’ identity privacy. These credentials adhere to the principle of minimal data disclosure and do not compromise any private information.

---

#### Algorithm 1 storeDoc

---

```

1: Input: string ipfsCID, string DID // User Data
2: struct Document
3:   string ipfsCID;
4:   uint256 timestamp; // Timestamp of order creation
5:   address public owner;
6:   mapping (string => Document) public didDocuments; //DID -> Document
7: function createDoc(string ipfsCID, string DID) public onlyOwner
8:   add Document(ipfsCID, timestamp) to didDocuments;
9: function getDetails(string DID) public view returns (Document)
10: return didDocuments[DID];

```

---

**Table 1.** DID document structure.

Parameter	Definition
context	W3C organization requirements
id	User’s DID
controller	Entity with the right to modify
verificationMethod	Structured data objects required for verification
authentication	Authorization signature
assertionMethod	Issue VC
capabilityDelegation	Authority delegation
keyAgreement	End-to-end communication key
service	Related services

---

#### 4.1.5. Registration Information Return

Government departments sign the VCs and subsequently transmit the signed VCs, along with the original data, back to the users through a secure channel, thereby ensuring their authenticity and integrity.

#### 4.2. Authentication

The core of the platform certification is to verify the content of the VC, specifically by verifying the digital signature of the VC provider to ascertain the authenticity of the VC.

The entire process is based on ECDSA [43]. During the signing phase, a random number  $k$  is chosen as the temporary key ( $0 < k < q$ ), and a point  $R$  on the elliptical curve is calculated, along with  $r$ , as shown in Equations (3) and (4). The secure hash function  $H(*)$  is used to calculate  $s$ , generating the signature for the message  $m$ , as shown in Equation (5).

$$R = kG = (x_R, y_R) \tag{3}$$

$$r = x_R \bmod q \tag{4}$$

$$s = k^{-1} (H(m) + r \cdot d) \bmod q \tag{5}$$

$(m, (r, s))$

The verifier uses  $(r, s)$ ,  $P_A$ , and  $H(m)$  to compute  $u_1$  and  $u_2$ , as well as the point  $P$ , as shown in Equation (6).

$$\begin{aligned} u_1 &= H(m)s^{-1} \bmod q \\ u_2 &= rs^{-1} \bmod q \\ P &= u_1G + u_2P_A \end{aligned} \tag{6}$$

The validity of the signature is verified by comparing the x-coordinate of  $P$  with  $r$ . Below is the entire certification process.

Users sign their authentication requests with their private key to prove that they are the legitimate holders of the decentralized identity. After receiving the requests, the platform performs two key verification operations: firstly, by checking the signature to confirm that the current requester is indeed the holder of the DID, and secondly, by verifying both the authenticity of the VC signature provided by the requester and the conformity of its content with the platform’s requirements, as shown in Figure 2. After completing user identity authentication, the platform categorizes users into drivers and passengers based on the attributes in the VC, providing them with respective attribute keys and other relevant data.

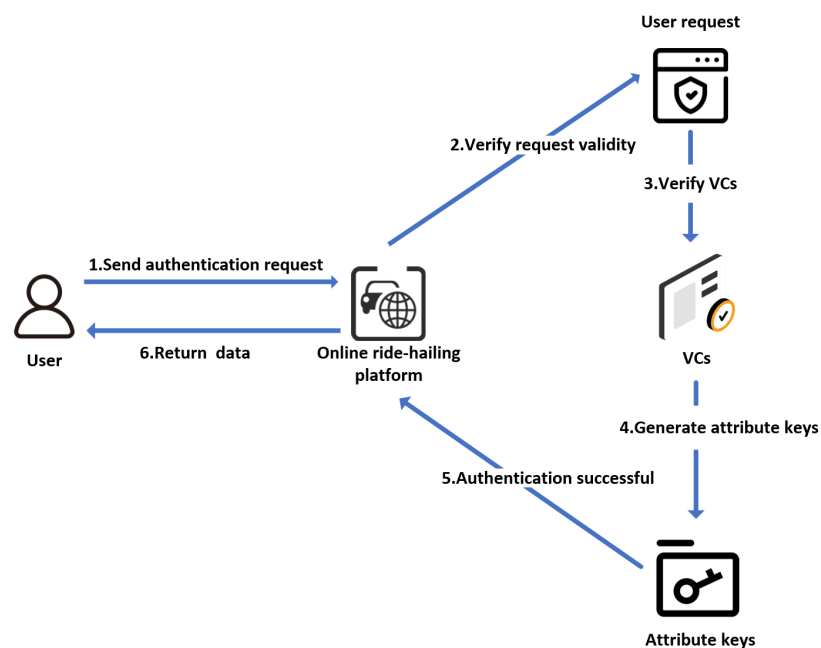


Figure 2. User authentication.

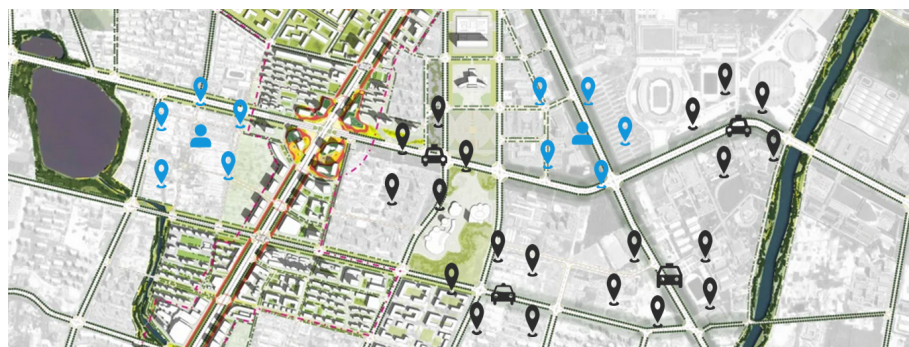
### 5. Ride-Hailing Platform Services

In the field of ride-hailing applications, location information contains a vast amount of potential value. To further enhance the protection of drivers’ and passengers’ personal privacy information while ensuring matching quality, the subsequent sections of this chapter will introduce an innovative ride-hailing matching service model.



### 5.1. Fuzzy Location Information Generation

When the ride-hailing service is initiated, it first obtains the users' approximate locations (such as city or district level) and displays nearby POI. The users then select POI sets from the platform's application as their current fuzzy locations, with blue representing the passenger sets and black representing the driver sets, as shown in Figure 3. In this process, it is assumed that the platform is honest and only obtains the users' fuzzy locations.



**Figure 3.** Fuzzy location selection.

### 5.2. Fuzzy Location Matching

Both passengers and drivers submit their selected sets of fuzzy points of interest,  $POI_p\{p1, p2, p3, \dots\}$  and  $POI_d\{d1, d2, d3, \dots\}$ , to the ride-hailing platform. The platform's matching algorithm maps POI coordinates in the sets into grid identifiers based on the city area where the provider is located. It then performs matching operations between drivers and passengers within the grid of the passenger's points of interest and its adjacent grids. Initially, the system initializes the hash functions required for MinHash signatures, with the hash functions of the form  $h(x) = (ax + b) \bmod c$ , where  $c$  is typically a large prime number. This hash function generates a series of different hash values for the subsequent calculation of MinHash signatures. The algorithm calculates multiple hashes for each element in each user's POI set and records the minimum hash result from all functions, forming the MinHash signature of that user's POI set. These MinHash signatures are then divided into several bands, each containing a certain number of consecutive hash values. Each band is hashed to produce a band hash value, which is used to index the users into corresponding LSH buckets. The algorithm then searches for drivers with the same LSH band hash values as the passenger and performs Jaccard similarity [44] matching on these drivers' MinHash signatures. Based on the similarity, the platform identifies the drivers who are closer to the passenger, as illustrated in Algorithm 2.

The matching method in this scheme specifically targets drivers within the grid of the passenger's interest point and its neighboring grids for precise matching, effectively preventing distant drivers from participating in irrelevant matching processes, thereby significantly enhancing the accuracy and quality of the matches. Additionally, in situations where driver resources are scarce, the system will flexibly solicit passenger opinions and appropriately expand the matching range to ensure that a relatively nearby driver can be found. This layered expansion strategy not only ensures the accuracy of the matches but also effectively mitigates the risk of matching failure that may arise due to insufficient driver numbers.

**Algorithm 2** fuzzyMatching

---

```

1: Input: Set<User> users // Users with their current POI
2: Output: drivers nearby
3: function indexUser(Set<User> users)
4:   for each user in users do
5:     for each band in user's MinHash signature do
6:       bandHash = Hash(band);
7:       add user to lshBucket according to bandHash;
8:     end for
9:   end for
10: function findPotentialMatches(User user)
11:   potentialMatches = empty set;
12:   for each band in user's MinHash signature do
13:     bandHash = Hash(band);
14:     get lshBucket by bandHash;
15:     for each candidate in lshBucket do
16:       if candidate != passenger // Not a passenger
17:         add candidate to potentialMatches;
18:       end if
19:     end for
20:   end for
21:   return potentialMatches;
22: function findMatchesForPassenger(User passenger)
23:   potentialMatches = findPotentialMatches(passenger);
24:   sort potentialMatches by similarity score;
25:   return nearby results;

```

---

## 5.3. Demand Matching and Order Generation

In the online ride-hailing business, “centralized dispatch” is a mainstream matching mode, but its matching results are completely determined by the platform algorithm. In order to improve the control of passengers over the order matching results, this solution adopts a hybrid encryption algorithm based on attributes. The passenger terminal provides access strategies to the platform to select drivers who meet the passenger's needs, thereby reducing the impact of the opacity of the platform-centralized matching algorithm on the matching results, reducing the platform's absolute control over the results and improving the security and fairness of matching.

## 5.3.1. Initialization

Generate the public parameters required for pairing  $\langle e, g, G_1, G_T, Z_r \rangle$ , as shown in Table 2. Using the public parameters, randomly generate  $\alpha, \beta \in Z_r$ , then compute  $g^\beta$ ,  $g^\alpha$ , and  $Y = e(g, g)^\alpha$ , where  $g$  is a generator of  $G_1$ . The system's public key is shown in Equation (7).

$$pk = (Y, g^\beta) \quad (7)$$

## 5.3.2. Key Generation

Select a random number  $t \in Z_r$  for calculating  $D = g^\alpha g^{t\beta}$  and  $D_0 = g^t$ . Here,  $D$  ensures randomness in the key, while  $D_0$  adds to the key's security and randomness. For each element in the user's attribute list, generate the corresponding  $D_i = H(i)^t$ . These elements together constitute the attribute private key, as shown in Equation (8).

$$\text{CP-ABE}_{sk} = [D, D_0, \{D_i \dots\}] \quad (8)$$

**Table 2.** Parameter definitions.

Parameter	Definition
$\alpha, \beta, r_n$	Random integers
$e, g$	The public parameters
$G1, GT$	Random group element
$Z_r$	Set of integers
$pk$	System’s public key
$M$	Plaintext message
$g^\alpha$	System’s master key
$D, D_0, \{D_i \dots\}$	Attribute private key
$S$	Source parameter on the group
$K_{AES}$	Symmetric key
$H(*)$	Attribute value hash mapping to group element
$U$	Secret key attribute collection
$T$	Access tree
$n$	Leaf node attributes
$ct$	Ciphertext
$\lambda_n, P_j$	Secret shard during Encryption and Decryption

### 5.3.3. Encryption

In order to perform encryption operations, the source parameter  $S \in G_T$  is first derived as a symmetric key, as shown in Equation (9), and the corresponding ciphertext information is generated, as shown in Equation (10). Then, select a random number  $s \in Z_r$  and compute  $C = S \cdot e(g, g)^{\alpha s}$  and  $C_0 = g^s$ . The secret  $s$  is divided along an access tree, ensuring that each leaf node’s attribute  $n$  corresponds to the secret shard  $\lambda_n$ . For each leaf node attribute, calculate  $C_n^1 = g^{\beta \lambda_n} H(n)^{-r_n}$ ,  $C_n^2 = g^{r_n}$ . The corresponding ciphertext is generated as shown in Equation (11).

$$K_{AES} = SHA256(S) \tag{9}$$

$$C_M = Enc_{K_{AES}}(M) \tag{10}$$

$$ct = (C_M, C, C_0, \{C_n^1, C_n^2 \dots\}) \tag{11}$$

### 5.3.4. Decryption

If the decryptor’s attribute set  $U$  satisfies the ciphertext access tree  $T$ , decryption can proceed. For the coinciding attributes  $j$  between  $U$  and the leaf node attribute set of  $T$ , compute  $P_j = e(C_j^1, D_0)e(C_j^2, D_j) = e(g, g)^{\beta t \lambda_j}$ . Start recursive operations from the root node, exponentiating the shard of each child node based on Lagrange interpolation factors and performing consecutive multiplications. Finally, recover the root node’s secret value in the form of  $e(g, g)^{\beta t s}$ . Subsequently, compute  $S$  from this value and reconstruct the AES key to decrypt the message, as shown in Equation (12).

$$\begin{aligned} e(g, g)^{\alpha s} &= e(C_0, D) / e(g, g)^{\beta t s} \\ S &= C / e(g, g)^{\alpha s} \\ M &= Dec_{K_{AES}}(C_M) \end{aligned} \tag{12}$$

In the preliminary phase, the ride-hailing platform initializes system parameters such as the public key  $pk$  and the master key  $g^\alpha$  and generates corresponding attribute private keys for drivers based on their vehicle type, service type, and affiliated service provider as specified in their VCs. At the same time, it is necessary to ensure that all drivers’ attribute keys are generated by the same  $g^\alpha$ . Passengers, based on their individual needs, select one or more attributes from three driver categories, and the passenger terminal automatically constructs an access tree policy based on these selections, securely submitting it to the platform.

The platform uses the policy to encrypt the passenger's DID and obtains the ciphertext, which is then sent to the drivers around the passenger based on the fuzzy matching results. This mechanism ensures that only drivers who meet the access policy and are close to the passenger can take the service request. Once the driver terminal successfully decrypts *ct*, it will actively request location information from the passenger. The passenger terminal will automatically perform a key exchange using the DID public key of the requesting driver to establish a shared secret. This shared secret will then be used to encrypt the location information entered by the passenger. The encrypted location ciphertext will be transmitted to the corresponding driver through the TLS/SSL secure channel. This process ensures that, even if the data passes through the platform's server, their content always remains encrypted and cannot be decrypted and accessed by unauthorized third parties. Finally, the drivers use the corresponding key to decrypt the location ciphertext and decide whether or not to accept the service request. Upon arrival at the passenger's location, the driver who accepted the ride request verifies the passenger's identity through the decryption of the DID from the *ct*.

The ride-hailing platform utilizes Algorithm 3 to generate transaction orders on the blockchain. To address potential emergencies that may arise during ride-hailing services, the orders record the DIDs of both drivers and passengers, stored as hash values on the blockchain. In emergency situations, the platform can provide the actual DIDs to regulatory agencies, which can then follow legal procedures to inquire about relevant identity information from government departments.

---

#### Algorithm 3 createOrder

---

```

1: Input: string ipfsCID // Order identifier
2: struct Order
3:   string ipfsCID;
4:   uint256 timestamp; // Timestamp of order creation
5: mapping(uint => Order) public orders; // Map of orders
6: function createOrder(string ipfsCID) public onlyOwner
7:   add Order(ipfsCID, currentTime) to orders;
8: function getOrder(uint orderId) public view returns (Order)
9:   return orders[orderId];

```

---

## 6. Performance Analysis

### 6.1. Security Analysis

This paper explores the application of blockchain, fuzzy matching, and data encryption technology in protecting the privacy data security of online ride-hailing users. The immutability and decentralization of blockchain build a secure and transparent digital environment. Based on such technology, the decentralized identity system built on blockchain gives users great autonomy and effectively avoids excessive collection and abuse of personal information. Fuzzy matching technology replaces the precise location with a set of points of interest to reduce the risk of inferring user identity based on location information. Data encryption stipulates the readability of data through algorithms and combines access policies to limit the access group, so that data is effectively protected during transmission and storage.

These technologies have brought innovative solutions to the field of online ride-hailing, but multiple factors still need to be considered in actual deployment. While blockchain and smart contract technologies have established an open and trusted environment, they inherently face challenges in safeguarding the privacy of user data. In addition, drivers are usually semi-honest, which means that while providing services, drivers may pursue their own interests or even snoop on user privacy. Therefore, this scheme deeply analyzes the above factors when designing and formulates corresponding countermeasures for the following threats that they may bring.

- **Threat:** Potential attackers can trace all transaction information of a specific user stored on the blockchain to analyze the user's location information, and then infer the user's personal privacy.  
**Resist Threat:** To better protect the privacy of users, both drivers and passengers provide a collection of POI representing their current locations to the platform. The passenger's encrypted location information will be sent to each eligible driver through end-to-end encryption technology. Only the driver who accepts the order can access the precise location of the passenger. The online ride-hailing platform can only obtain fuzzy information about the starting point for ride matching but cannot obtain any data related to the destination. This design ensures that transactions stored on the blockchain by the platform are not linked to the user's complete travel information, thereby effectively preventing potential attackers from analyzing user privacy through on-chain transaction orders.
- **Threat:** Drivers might use multiple platform accounts to await incoming orders, then select the most advantageous ride request for themselves.  
**Resist Threat:** By using DIDs for authentication, the uniqueness and immutability of a DID ensure that each driver can only have one DID associated with their identity. In order to address the problem of drivers authenticating on multiple online ride-hailing platforms to increase the chance of receiving orders, this scheme defines the driver's current status within the "service" field in the DID document. Once the driver starts accepting orders on any platform, the status in serviceEndpoint will change from 'available' to 'busy'. All platforms verify the driver's current status before accepting orders, and only drivers marked as "available" can accept orders. This effectively prevents drivers from using multiple accounts to accept more orders, thus enhancing service quality.
- **Threat:** If the passenger's encrypted location only includes precise coordinates, it would enable all drivers who meet the specified attributes to access the exact location, potentially increasing the security risks for the passenger during the pick-up and drop-off processes.  
**Resist Threat:** Considering user experience for both parties, this scheme matches drivers near passengers based on fuzzy location information. Subsequently, it identifies a suitable group of drivers according to the requirements provided by the passengers and provides the passengers' encrypted location data. Drivers decide whether or not to accept the order based on the preliminary decrypted POI of the passenger's destination. Critically, this process ensures that both parties have autonomy in their choices, while guaranteeing that only the driver who ultimately accepts the order receives the passenger's precise location information.

## 6.2. Experimental Analysis

To evaluate the performance of the proposed scheme in terms of security and effectiveness, decentralized identity generation is conducted on a 64-bit Windows PC (Lenovo, Beijing, China) equipped with 16 GB of memory and a 3 GHz AMD R9-7945 CPU. The complete process of using this identity for authentication and ride matching on a ride-hailing platform is simulated. In the experimental setup, Ganache (version 2.7.1) is used for setting up a test chain, while smart contract testing is carried out in Remix (version 0.54.0) and Visual Studio Code (version 1.93.0). Off-chain operations are entirely implemented in Java (version 17.0.7) and simulated in the IDEA (version 2021.1.3) editor.

In the entire decentralized identity generation process, users generate two sets of ECC key pairs. One set is for digital signing and DID creation, and another set is for backup in potential recovery operations. This ensures continuity of identity management by providing a fallback mechanism in the event of primary private key loss, as detailed in Figure 4.

Public key generation: 3056301006072a8648ce3d020106052b8104000a03420004235db6138644688f035d88ca5380abb830c  
 Private key generation: 30818d020100301006072a8648ce3d020106052b8104000a0476307402010104204460a0daa6ebd93f  
 Alternate public key generation: 3056301006072a8648ce3d020106052b8104000a03420004cf2c0da4c7ede06de13e406ff  
 Alternate private key generation: 30818d020100301006072a8648ce3d020106052b8104000a047630740201010420111b27

Figure 4. Key pair generation.

This scheme simulates the process of users generating DIDs and government authorities creating corresponding DID documents for users according to the W3C’s decentralized identifier specification and storing the data on the blockchain during its design. These documents’ CID are then stored on the blockchain, as illustrated in Figure 5. Figure 6 shows the content of the DID document in IPFS corresponding to the CID stored on the blockchain. The DID documents are jointly controlled by users and government departments, ensuring the authority of the data. Meanwhile, the scheme grants users full control over the validation methods for DID documents, allowing them to manage and update public key information based on actual demands.

Table 3 shows the time taken for each stage. The data show that the service costs provided by government departments and ride-hailing platforms are relatively low, while the time taken for users to generate key pairs locally is the longest. Nevertheless, the total duration of the process does not exceed 0.3 s, fully meeting the usage needs.

Table 3. Decentralized identity-related operations.

Executor	Pattern	Operations	Time (ms)
Government department	off-chain	DID document generation	2
	off-chain	Store to IPFS	70
	on-chain	Store CID	26
Online ride-hailing platform	off-chain	Authentication	5
User	off-chain	Key generation	140
	off-chain	DID generation	12
	off-chain	Sign	3

The experiment separately calculates the time consumption when varying numbers of drivers surrounding the grid where the passenger resides engage in fuzzy location matching, and subsequently generates matching results ordered by proximity. Figure 7 shows the results stored in IPFS.

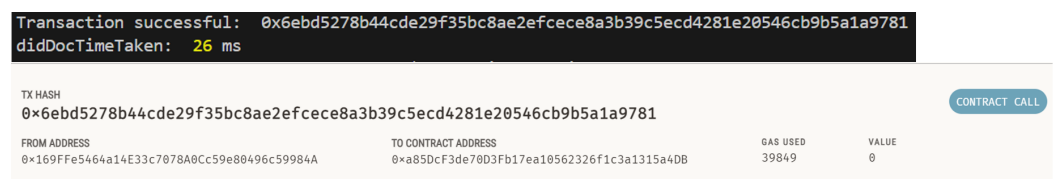


Figure 5. DID document generation.

As the number of participating drivers increases, there will be more drivers near the passenger who can obtain the ciphertext *ct*, thereby increasing the probability of generating orders. Multiple experiments yielded the average time overhead for each group, as shown in Figure 8. The results indicate that when the number of participating drivers reaches 100, the matching time does not exceed 7 ms, fully meeting the system’s requirements. Additionally, if this technique is implemented in a distributed server architecture, where each node manages the matching process within its region, the advantages of MinHash and LSH will be more fully realized, further enhancing the system’s efficiency and robustness.

ipfs / QmZW...hMbw

```
{
  "@context": [
    "https://www.w3.org/ns/did/v1"
  ],
  "id": "did:Govchain:2FTvA4Ymhbr9U8cnPa1hEGygAcpi",
  "controller": [
    "did:Govchain:2FTvA4Ymhbr9U8cnPa1hEGygAcpi",
    "did:Government:3JmpRyxAohiw9JTt7SQvtLYrpFry"
  ],
  "verificationMethod": [
    {
      "id": "did:Govchain:2FTvA4Ymhbr9U8cnPa1hEGygAcpi#public-key-Hex-1",
      "type": "EcdsaSecp256k1VerificationKey2019",
      "controller": "did:Govchain:2FTvA4Ymhbr9U8cnPa1hEGygAcpi",
      "publicKeyHex": "3056301006072a8648ce3d020106052b8104000a03420004235db6138644688f03"
    },
    {
      "id": "did:Govchain:2FTvA4Ymhbr9U8cnPa1hEGygAcpi#public-key-Hex-2",
      "type": "X25519KeyAgreementKey2019",
      "controller": "did:Govchain:2FTvA4Ymhbr9U8cnPa1hEGygAcpi",
      "publicKeyHex": "7071F5C5000BAD76E4584DAB83E3ED64164BF28D965F6BD7A8B38FA55317A11"
    }
  ],
  "authentication": [
    "did:Govchain:2FTvA4Ymhbr9U8cnPa1hEGygAcpi#public-key-Hex-1"
  ],
  "assertionMethod": [
    "did:Govchain:2FTvA4Ymhbr9U8cnPa1hEGygAcpi#public-key-Hex-1"
  ],
  "capabilityDelegation": [
    "did:Govchain:2FTvA4Ymhbr9U8cnPa1hEGygAcpi#public-key-Hex-1"
  ],
  "keyAgreement": [
    "did:Govchain:2FTvA4Ymhbr9U8cnPa1hEGygAcpi#public-key-Hex-2"
  ],
  "service": [
    {
      "id": "did:Govchain:2FTvA4Ymhbr9U8cnPa1hEGygAcpi#edv",
      "type": "EncryptedDataVault",
      "serviceEndpoint": "https://edv.example.com/"
    },
    {
      "id": "did:Govchain:2FTvA4Ymhbr9U8cnPa1hEGygAcpi#status",
      "type": "StatusService",
      "serviceEndpoint": "https://status.example.com/"
    }
  ]
}
```

Figure 6. DID and DID document.

ipfs / QmVM...HB2W

ipfs / QmXD...FJJZ

```
Driver3
Driver11
Driver9
Driver13
Driver6
Driver4
Driver18
Driver17
```

```
Driver3
Driver27
Driver21
Driver11
Driver9
Driver26
Driver13
Driver29
Driver22
Driver25
Driver24
Driver30
```

Figure 7. Match result.

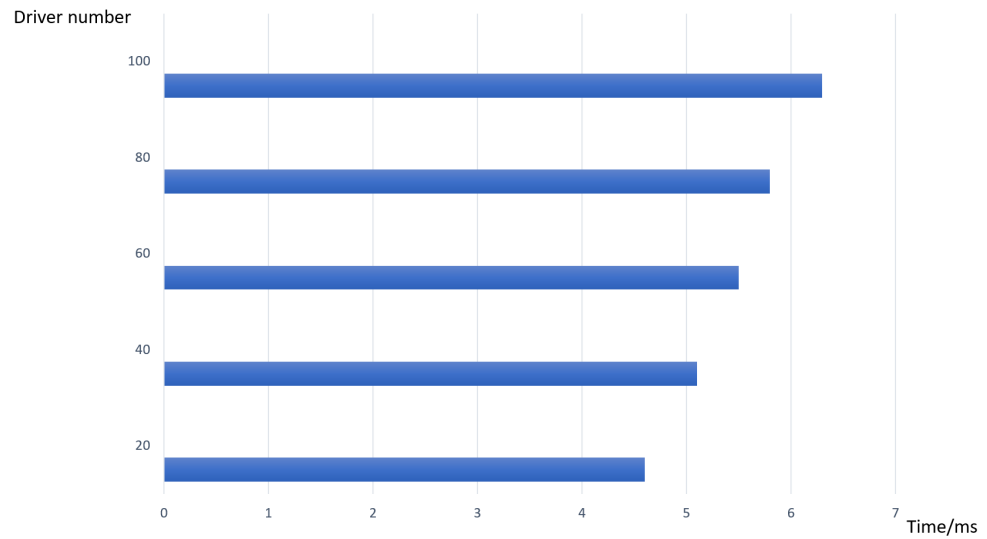


Figure 8. Match processing time.

During the execution of the hybrid encryption algorithm, the passenger terminal needs to generate more suitable access control policies based on passenger requirements in order to provide better services. Considering that there are three types of driver attribute, with the optional attributes usually not exceeding 15, the number of leaf nodes of the access tree in this scheme is set between 10 and 15. Figure 9 shows the execution time of each component in the hybrid encryption process. To ensure the security of user data and achieve attribute-based fine-grained access control, the encryption process becomes the primary time-consuming part of the algorithm. When the number of attribute nodes is 15, the total execution time of the hybrid encryption algorithm does not exceed 1.2 s, and the decryption operation takes no more than 0.1s, fully meeting the needs of ride-hailing services.

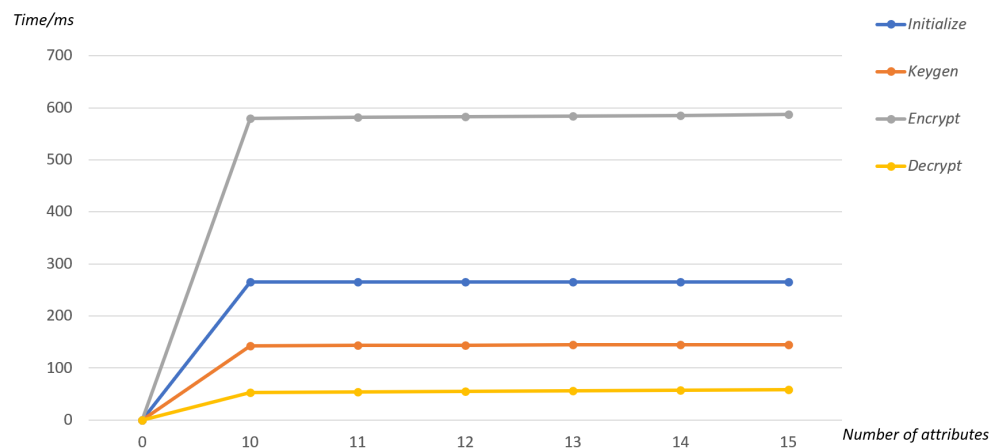


Figure 9. Hybrid encryption algorithm overhead.

### 6.3. Performance Comparison

This scheme has undergone a comprehensive comparative analysis with other privacy protection schemes, focusing on the evaluation metrics outlined in Table 4.



**Table 4.** Comparison of schemes.

Scheme	Protect User (Driver & Passenger) Privacy	User Anonymity	User Autonomy	Emergency Response	Complete Decoupling of Key Services
Fadhil et al. [9]	yes	no	no	yes	no
Sánchez et al. [17]	yes	no	no	yes	no
Pham et al. [19]	no	yes	no	no	no
Shen et al. [21]	yes	yes	no	no	no
Pham et al. [22]	yes	no	no	yes	no
Our scheme	yes	yes	yes	yes	yes

## 7. Analysis and Discussion

This scheme achieves the decoupling of key services in ride-hailing, enabling users to utilize decentralized identities for authentication on ride-hailing platforms while enjoying the services without providing precise location information to the platform. This approach prevents the association of identity data with location-based trip data within the platform, thereby maximizing user privacy protection.

The process involves multiple stakeholders, including government departments, drivers, passengers, ride-hailing platforms, and regulatory agencies. Each party plays a crucial role in the ride-hailing ecosystem, taking on specific responsibilities and challenges. The following paragraphs will systematically analyze key issues related to user privacy, enterprise risk management, and regulatory considerations.

In this scheme, users assume two main roles. First, as holders of decentralized identities, users are empowered through DID systems by gaining greater control over their personal information. As identity holders, they can selectively disclose data and adjust privacy settings to meet their individual needs. Although users may have limited direct influence over the design and implementation of the service, the transparency of privacy settings and the level of control provided offer them significant autonomy. Additionally, users will receive clear guidance on privacy protection to ensure they can understand and choose the options that best suit their needs. Second, as users of ride-hailing platforms, especially passengers, they can designate vague locations and vehicle preferences through an intuitive interface. When combined with the on-demand matching method of this solution, it not only safeguards their privacy but also grants them appropriate choices and control.

Adopting the DID framework requires ride-hailing service providers to establish robust Enterprise Risk Management (ERM) strategies. ERM enables these companies to systematically identify, assess, and mitigate risks related to data privacy, technology, compliance, and third-party interactions. Moreover, the platform must ensure that its system complies with relevant legal regulations, such as the GDPR and CCPA. While decentralized technology inherently reduces some risks by minimizing centralized data storage, it also introduces new challenges, such as unauthorized access and data breaches. To address these issues, companies must implement comprehensive strategies, including data encryption, zero-knowledge proofs, and continuous risk monitoring. A dedicated risk management team plays a crucial role in ensuring these strategies are effectively executed, maintaining platform stability, and preserving user trust.

The transition to a decentralized identity system also presents significant regulatory challenges. Government and regulatory agencies must address the complexities of accountability and data traceability in decentralized networks. Ensuring data security and audit feasibility requires clear guidelines and effective traceability mechanisms. Additionally, the intersection of technology within ride-hailing services and real-world safety demands regulatory oversight to protect both drivers and passengers. This includes establishing crime prevention measures, ensuring swift responses to emergencies, and preventing the

misuse of technology. Regulatory authorities are responsible for defining and monitoring the platform's obligations to safeguard user safety and information protection, thereby creating a secure and reliable environment for all stakeholders.

## 8. Conclusions

This paper proposes a privacy protection scheme for ride-hailing services based on decentralized identity, which aims to completely decouple user identity registration services from location-based ride services within ride-hailing platforms. The scheme restricts the power of ride-hailing platforms to prevent them from excessively acquiring and abusing users' private information. In this scheme, identity services are handled by government departments, and users use decentralized identities obtained from government departments for authentication on ride-hailing platforms. This approach decouples identity registration services from the platforms, thereby preventing the platforms from accessing user-related identity privacy information. To further ensure that the ride-hailing platform does not have access to users' precise location information, the scheme integrates fuzzy location matching technology and ciphertext-policy attribute-based hybrid encryption algorithms, ensuring that ride-hailing platforms only obtain users' current fuzzy location data for real-time matching purposes, restricting access to location information. The scheme effectively addresses privacy leakage concerns faced by users during the order generation process, fully safeguarding user privacy. Regulatory authorities have been introduced to handle emergencies arising during ride-hailing services, leveraging the advantages of decentralized identities to better protect user safety. This scheme establishes a mutually trusted privacy protection framework among users, government departments, and ride-hailing platforms, achieving the complete decoupling of platform services and enhancing the security and reliability of ride-hailing services. Future work will focus on improving the system and developing a demonstration project to further enhance the scheme's efficiency, thereby better promoting the application of decentralized identity to more effectively protect user privacy.

**Author Contributions:** N.S. was the advisor. Y.L. (Yuxuan Liu) designed the scheme. Y.L. (Yuxuan Liu) carried out the implementation. Y.L. (Yuxuan Liu) wrote the manuscript. N.S., Y.Z. and Y.L. (Yining Liu) revised the final version of the text. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no grant or funding from any funding agency in the public, commercial, or not-for-profit sectors.

**Data Availability Statement:** All data are contained within the article.

**Conflicts of Interest:** Author Yuanyi Zhang were employed by the company Shanghai Shentie Information Engineering Co., Ltd. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

## References

1. Ma, N.F.; Hanrahan, B.V. Unpacking sharing in the peer-to-peer economy: The impact of shared needs and backgrounds on ride-sharing. *Proc. ACM Hum.-Comput. Interact.* **2020**, *4*, 1–19. [CrossRef]
2. Zhang, W.; Zhong, S. Data Legal Supervision of Online Car-Hailing Platform Based on Big Data Technology and Edge Computing. *Wirel. Commun. Mob. Comput.* **2022**, *2022*, 5298152. [CrossRef]
3. Tibken, S. Uber Accused of Tracking Celebs, Politicians. Available online: <https://www.cnet.com/news/privacy/uber-lawsuit-alleges-startup-tracked-celebs-politicians/> (accessed on 18 September 2024).
4. Canivel, R.S.C. Uber PH Confirms Data of Filipino Users among Those Hacked— NPC. Available online: <https://technology.inquirer.net/69763/breaking-internet-hacking-uber-national-privacy-commission-breach-personal-information/> (accessed on 18 September 2024).
5. Chow, C.Y.; Mokbel, M.F. Enabling private continuous queries for revealed user locations. In *International Symposium on Spatial and Temporal Databases*; Springer: Berlin/Heidelberg, Germany, 2007; pp. 258–275.
6. Hengartner, U. Hiding location information from location-based services. In *Proceedings of the 2007 International Conference on Mobile Data Management*, Mannheim, Germany, 1 May 2007; IEEE: Piscataway, NJ, USA, 2007; pp. 268–272.

7. Houerbi, K.R.; Machfar, D.; Ayed, H.K.B. Blockchain for Ridesharing: A Systematic Literature Review. In Proceedings of the 2023 International Conference on Smart Applications, Communications and Networking (SmartNets), Istanbul, Turkiye, 25–27 July 2023; IEEE: Piscataway, NJ, USA, 2023; pp. 1–6.
8. Renu, S.A.; Banik, B.G. Implementation of a secure ridesharing DApp using smart contracts on Ethereum blockchain. *Int. J. Saf. Secur. Eng.* **2021**, *11*, 167–173.
9. Fadhil, M.; Sukarno, P.; Wardana, A.A. Decentralized Privacy-Preserving Solution Through Blockchain Smart Contracts and Spatial Cloaking for Ride Sharing Application. In *Science and Information Conference*; Springer: Cham, Switzerland, 2024; pp. 377–395.
10. Shawon, S.K.; Ahammad, H.; Shetu, S.Z.; Rahman, M.; Hossain, S.A. DIUcerts DApp: A blockchain-based solution for verification of educational certificates. In Proceedings of the 2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT), Kharagpur, India, 6–8 July 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 1–10.
11. Gulati, H.; Huang, C.T. Self-sovereign dynamic digital identities based on blockchain technology. In Proceedings of the 2019 SoutheastCon, Huntsville, AL, USA, 11–14 April 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 1–6.
12. Salem, S.H.G.; Hassan, A.Y.; Moustafa, M.S.; Hassan, M.N. Blockchain-based biometric identity management. *Clust. Comput.* **2024**, *27*, 3741–3752. [[CrossRef](#)]
13. Feher, K. Digital identity and the online self: Footprint strategies—An exploratory and comparative research study. *J. Inf. Sci.* **2021**, *47*, 192–205. [[CrossRef](#)]
14. Haque, A.B.; Bhushan, B.; Dhiman, G. Conceptualizing smart city applications: Requirements, architecture, security issues, and emerging trends. *Expert Syst.* **2022**, *39*, e12753. [[CrossRef](#)]
15. Wang, F.; De Filippi, P. Self-sovereign identity in a globalized world: Credentials-based identity systems as a driver for economic inclusion. *Front. Blockchain* **2020**, *2*, 28. [[CrossRef](#)]
16. Wang, Y.; Su, Z.; Ni, J.; Zhang, N.; Shen, X. Blockchain-empowered space-air-ground integrated networks: Opportunities, challenges, and solutions. *IEEE Commun. Surv. Tutorials* **2021**, *24*, 160–209. [[CrossRef](#)]
17. Sánchez, D.; Martínez, S.; Domingo-Ferrer, J. Co-utile P2P ridesharing via decentralization and reputation management. *Transp. Res. Part Emerg. Technol.* **2016**, *73*, 147–166. [[CrossRef](#)]
18. Kang, J.; Yu, R.; Huang, X.; Zhang, Y. Privacy-preserved pseudonym scheme for fog computing supported internet of vehicles. *IEEE Trans. Intell. Transp. Syst.* **2017**, *19*, 2627–2637. [[CrossRef](#)]
19. Pham, A.; Dacosta, I.; Jacot-Guillarmod, B.; Huguenin, K.; Hajar, T.; Tramèr, F.; Gligor, V.; Hubaux, J.P. Privateride: A privacy-enhanced ride-hailing service. *Proc. Priv. Enhancing Technol.* **2017**, *2017*, 38–56. [[CrossRef](#)]
20. Hong, Y.; Yang, L.; Xiong, Z.; Kanhere, S.S.; Jiang, H. OCHJRNCHAIN: A blockchain-based security data sharing framework for online car-hailing journey. *IEEE Trans. Intell. Transp. Syst.* **2024**, *25*, 5299–5311. [[CrossRef](#)]
21. Shen, X.; Wang, Z.; Wang, B.; Wang, L.; Pei, Q. A Privacy-Preserving Ride-Matching Scheme Without a Trusted Third-Party Server. *IEEE Syst. J.* **2023**, *17*, 6413–6424. [[CrossRef](#)]
22. Pham, A.; Dacosta, I.; Endignoux, G.; Pastoriza, J.R.T.; Huguenin, K.; Hubaux, J.P. ORide: A Privacy-Preserving yet Accountable Ride-Hailing Service. In Proceedings of the 26th USENIX Security Symposium (USENIX Security 17), Vancouver, BC, Canada, 16–18 August 2017; pp. 1235–1252.
23. Maram, D.; Malvai, H.; Zhang, F.; Jean-Louis, N.; Frolov, A.; Kell, T.; Lobban, T.; Moy, C.; Juels, A.; Miller, A. Candid: Can-do decentralized identity with legacy compatibility, sybil-resistance, and accountability. In Proceedings of the 2021 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, 24–27 May 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 1348–1366.
24. Kang, M.; Lemieux, V. A decentralized identity-based blockchain solution for privacy-preserving licensing of individual-controlled data to prevent unauthorized secondary data usage. *Ledger* **2021**, *6*, 126–151. [[CrossRef](#)]
25. Javed, I.T.; Alharbi, F.; Bellaj, B.; Margaria, T.; Crespi, N.; Qureshi, K.N. Health-ID: A blockchain-based decentralized identity management for remote healthcare. *Healthcare* **2021**, *9*, 712. [[CrossRef](#)]
26. Stockburger, L.; Kokosioulis, G.; Mukkamala, A.; Mukkamala, R.R.; Avital, M. Blockchain-enabled decentralized identity management: The case of self-sovereign identity in public transportation. *Blockchain Res. Appl.* **2021**, *2*, 100014. [[CrossRef](#)]
27. Nakamoto, S.; Bitcoin, A. A Peer-To-Peer Electronic Cash System. *Bitcoin* **2008**, *4*, 15. Available online: <https://bitcoin.org/bitcoin.pdf> (accessed on 18 September 2024).
28. Ali, V.; Norman, A.A.; Azzuhri, S.R.B. Characteristics of blockchain and its relationship with trust. *IEEE Access* **2023**, *11*, 15364–15374. [[CrossRef](#)]
29. Buterin, V. A next-generation smart contract and decentralized application platform. *White Paper* **2014**, *3*, 2-1.
30. Dutta, P.; Choi, T.M.; Somani, S.; Butala, R. Blockchain technology in supply chain operations: Applications, challenges and research opportunities. *Transp. Res. Part Logist. Transp. Rev.* **2020**, *142*, 102067. [[CrossRef](#)]
31. Raja Santhi, A.; Muthuswamy, P. Influence of blockchain technology in manufacturing supply chain and logistics. *Logistics* **2022**, *6*, 15. [[CrossRef](#)]
32. Brunner, C.; Gellersdörfer, U.; Knirsch, F.; Engel, D.; Matthes, F. Did and vc: Untangling decentralized identifiers and verifiable credentials for the web of trust. In Proceedings of the 2020 3rd International Conference on Blockchain Technology and Applications, Xi’an China, 14–16 December 2020; pp. 61–66.
33. W3C. DID Specification Registries. Available online: <https://www.w3.org/TR/did-spec-registries/> (accessed on 18 September 2024).
34. W3C. Decentralized Identifiers (DIDs) v1.0. Available online: <https://www.w3.org/TR/did-core/> (accessed on 18 September 2024).

35. W3C. Verifiable Credentials Data Model v1.1. Available online: <https://www.w3.org/TR/vc-data-model/> (accessed on 18 September 2024).
36. Broder, A.Z.; Charikar, M.; Frieze, A.M.; Mitzenmacher, M. Min-wise independent permutations. *J. Comput. Syst. Sci.* **1998**, *60*, 327–336. [[CrossRef](#)]
37. Indyk, P.; Motwani, R. Approximate nearest neighbors: Towards removing the curse of dimensionality. In Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing, Dallas, TX, USA, 24–26 May 1998; pp. 604–613.
38. Bethencourt, J.; Sahai, A.; Waters, B. Ciphertext-policy attribute-based encryption. In Proceedings of the 2007 IEEE Symposium on Security and Privacy (SP'07), Berkeley, CA, USA, 20–23 May 2007; IEEE: Piscataway, NJ, USA, 2007; pp. 321–334.
39. Abdullah, A.M. Advanced encryption standard (AES) algorithm to encrypt and decrypt data. *Cryptogr. Netw. Secur.* **2017**, *16*, 11.
40. Gruteser, M.; Grunwald, D. Anonymous usage of location-based services through spatial and temporal cloaking. In Proceedings of the 1st International Conference on Mobile Systems, Applications and Services, San Francisco, CA, USA, 5–8 May 2003; pp. 31–42.
41. Gedik, B.; Liu, L. Protecting location privacy with personalized k-anonymity: Architecture and algorithms. *IEEE Trans. Mob. Comput.* **2007**, *7*, 1–18. [[CrossRef](#)]
42. Fang, X.; Wu, Y. Investigation into the elliptic curve cryptography. In Proceedings of the 2017 3rd International Conference on Information Management (ICIM), Chengdu, China, 21–23 April 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 412–415.
43. Kobitz, N. An elliptic curve implementation of the finite field digital signature algorithm. In Proceedings of the Advances in Cryptology—CRYPTO'98: 18th Annual International Cryptology Conference Santa Barbara, CA, USA, 23–27 August 1998; Proceedings 18. Springer: Berlin, Germany, 1998; pp. 327–337.
44. Niwattanakul, S.; Singthongchai, J.; Naenudorn, E.; Wanapu, S. Using of Jaccard coefficient for keywords similarity. In Proceedings of the International Multiconference of Engineers and Computer Scientists, Hongkong, China, 13–15 March 2013; Volume 1, pp. 380–384.

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.