Hindawi
Mobile Information Systems
Volume 2021, Article ID 9997887, 15 pages
https://doi.org/10.1155/2021/9997887

Research Article

Blockchain-Enabled Privacy-Preserving Location Sharing Scheme for LBSNs

Liang Zhu, Xiaowei Liu, Liping Yu, Zengyu Cai, and Jianwei Zhang

School of Computer and Communication Engineering, Zhengzhou University of Light Industry, Zhengzhou 450001, China

Correspondence should be addressed to Liang Zhu; lzhu@zzuli.edu.cn

Received 17 March 2021; Revised 14 May 2021; Accepted 22 June 2021; Published 1 July 2021

Academic Editor: Xiaohong Jiang

The rise of Internet of Things (IoT) technology promotes the rapid development of location services industry. The idea of smart connectivity also provides a new direction for Location-Based Social Networks (LBSNs). However, due to limited calculate ability and internal storage space of IoT devices, historical location data of users is generally stored in the central server, which is likely to cause the disclosure of users' private data. In this paper, we propose a Blockchain-enabled Privacy-Preserving Location Sharing (B-PPLS) scheme, which is a new framework that not only protects user location privacy but also provides effective location sharing services for users. For B-PPLS, location data owners can share the location area instead of location coordinates to Requesters, in order to realize the location privacy preserving. Also, the Merkle hash tree is utilized to divide the location area, so as to realize the multilevel privacy preserving. Furthermore, four algorithms are proposed to achieve the four stages of initialization, location record, location sharing, and location verification, respectively. Finally, we analyze the security of the proposed B-PPLS scheme and compare the performance with other related location privacy-preserving schemes by experimental evaluation.

1. Introduction

The characteristic of Location-Based Social Networks (i.e., LBSNs) is that people can make use of "check-in" to achieve the sharing and propagation of location-based services in the virtual world [1]. With the incredible development of IoT technology and the growing popularity of mobile devices, there is a widespread need for LBSNs in social life, medical, military, and other areas [2]. Large numbers of applications are also developed, such as route navigation, friend discovery, and POI recommendation. However, it usually requires the real-time location data sharing between devices for IoT technology and LBSNs, which could potentially lead to serious breaches of users' privacy. In order to achieve location privacy preserving and provide personalized services for users, the architecture of Blockchain-enabled LBSNs (i.e., B-LBSNs) is designed in this paper.

As shown in Figure 1, three relational graphs (i.e., ① *user-location*, ② *user-user*, and ③ *location-location*) are generated in B-LBSNs. The user-location graph reflects the relation between the user and location, in which different locations are visited by different users. The user-user graph represents the relationship among users, which is built by utilizing Blockchain technology. Therefore, the identity of users is authentic, and the information interaction between users is secure. The location-location graph represents the relationship among geographical locations according to semantical information.

LBSNs refer to obtaining location information of terminal equipment through a variety of external positioning technologies, such as Global System for Mobile Communications (i.e., GSM), Code Division Multiple Access (i.e., CDMA), or Global Positioning System (i.e., GPS), which enable users to share location information and provide users with location-related services. The commonly used location information sharing mechanism is implemented through a third-party central server, which also means that a large amount of user privacy location information is received and managed by the central server. Firstly, it is difficult for the regulatory technique to access the centrally managed

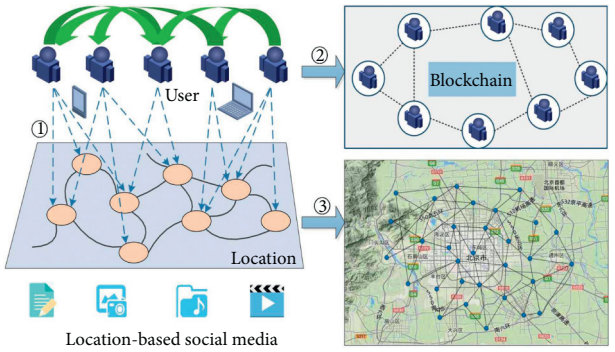

Figure 1: The architecture of B-LBSNs.

location data, and the lack of openness of location data leads to the security threats of illegal use and disclosure of location data because location information consists of a massive quantity of private information, such as work location, family information, and behavioral preferences. Once location information is obtained illegally, users' privacy and even personal safety will be seriously threatened. Secondly, the management approach makes it difficult to share location data between different third-party service requests. As the benefits of information sharing rapidly emerge, the phenomenon that various application services are allowed to access information between each other is becoming more and more common. For example, the location information of users is uploaded to LBSNs' server. Because the LBSNs' server has full control over the location information of the mobile users, the server may tamper with the data without the users' authorization. Various application servers cannot distinguish the authenticity of the location data returned by the LBSNs' server. Once the tampered information is received, the output results will be biased. If there is an issue of illegal manipulation of location data in the emergency first aid or defense, it is bound to have unimaginable consequences for people's physical health, daily life, or national security. As a result, the central server collects a huge amount of user-sensitive data, and users lose control over the location data stored in the centralized server.

Blockchain is a new application model of distributed data storage, which is essentially the decentralized distributed database [3]. If the Blockchain is applied to the location record storage of users, its distributed and decentralized characteristics can ensure that the user's location will not be stored and recorded separately by the third party's centralized service nodes. At the same time, Blockchain can ensure the immutability and nonrepudiation of the recorded location data by making use of the cryptographic mechanism. If the location information is safely stored in the Blockchain, it can eliminate the problems of illegal location tampering, illegal use, and illegal leakage caused by the centralized management of the third party. Also, it facilitates the secure location sharing among different nodes in the LBSNs. If necessary access control, privacy protection, and other measures are taken for the location information in the Blockchain, the location data can be controlled by the users. Therefore, the inherent characteristics of Blockchain technology enable it to store location data and realize the possibility of secure location sharing.

Blockchain technology not only realizes the safe sharing of location data but also increases the controllability of users to their own location data. Although its inherent characteristics can eliminate the security problems caused by centralized storage of location data, there are still some security threats described as following:

(1) Due to the open characteristics of Blockchain, any node in the LBSNs can have unauthorized access to the location data stored on the Blockchain if the confidentiality cannot be guaranteed. At this time, it will cause the user's location information to be fully disclosed, so as to bring huge security risks to the user.

(2) If the privacy protection of location information cannot be provided in the location sharing scheme, the malicious location requestor will infer the user's behavior habits and lifestyle based on the location information requested for many times [4]. It may cause the user to be tracked maliciously and personal attacked and other serious consequences.

(3) The malicious user may return fake location information to the location Requester. If the location requester cannot verify the authenticity of the user's location data, the forged location information will be regarded as the user's real location. It may lead to varying degrees of location-based analysis bias, service bias, and other problems.

(4) Once the content is uploaded to the Blockchain, it cannot be modified due to the immutability of the Blockchain. After the location information is uploaded with privacy protection, if the original location information cannot be completely recovered, the quality of the original location information will be permanently lost. In the cases where precise location is needed, such as medical institutions or police agencies, the user's original location cannot be obtained through Blockchain.

In this paper, we focus on Blockchain-enabled Privacy-Preserving Location Sharing (B-PPLS), which is a new framework that not only protects user location privacy but also provides effective location sharing services for users. The contributions of our work can be divided into three aspects as follows:

(1) We propose a Blockchain-enabled Privacy-Preserving Location Sharing scheme, named B-PPLS. In B-PPLS, users are able to share the location area instead of location coordinates to location Requesters, in order to realize the location privacy preserving. Also, the Merkle hash tree is utilized to divide the location area, so as to realize the multilevel privacy preserving.

(2) We present four algorithms to complete the processes of initialization, location record, location sharing, and location verification. Furthermore, we

make the security analysis of the B-PPLS scheme according to the proposed security objectives.

(3) We implement the designed algorithms and through the experiments to verify the safety and feasibility of B-PPLS.

The framework of this paper is organized as follows. Section 2 discusses the related work. Section 3 provides the overview of the B-PPLS scheme. In Section 4, we present the scheme designs of B-PPLS including initialization, location record, location sharing, and location verification. In Section 5, we give the security analysis of the B-PPLS scheme. Experiments and evaluations are discussed in Section 6. Finally, we conclude the paper and give future research directions in Section 7.

2. Related Work

In this section, the related research is conducted in three aspects. Firstly, the development process about location privacy preserving is explained. The second, the references about Blockchain-based location privacy preserving are described. At last, the works on encryption algorithms are researched.

2.1. Location Privacy Preserving. It involves the research on location privacy preserving when users share their location information to location requesters. The main methods of location privacy preserving are false location, k-anonymity, differential privacy, and cryptography. The method of false location is to realize the location privacy preserving by returning false location. The method of false location which owns high cost is not suitable for mobile devices because of the constrained resource. In order to reduce the cost, Liu et al. [5] utilized the Bayesian games to improve the dummy generation. It can help users achieve optimized payoffs. The two-tier schema based on k-anonymity was proposed by Fei et al. [6] to reduce the privacy-preserving cost. Pingley et al. [7] proposed a disturbance method based on the Hibert curve, in which the user added noise to the real position to get the false position. This solution is the first end-to-end solution that considers the quality of communication service while protecting location privacy and improving LBS accuracy. k-anonymity is the popular technique for location privacy preserving. Also, the incremental clique-based cloaking algorithm, which considered k-anonymity and cloaking granularity, was proposed by Pan et al. [8]. In order to achieve the personalized privacy preserving, Gedik and Liu [9] proposed a flexible privacy-preserving framework based on location k-anonymity. Although the location obfuscation method can protect location privacy, the data utility was lower. The differential-and-distortion framework was designed by Wang et al. [10] to reduce the data loss during the process of location obfuscation. The differential privacy model can achieve the higher data utility for privacy preserving. Xu et al. [11] proposed the DP-LTOD scheme, which obfuscated original trajectory sequences into differential privacy-guaranteed trajectory sequences for privacy preserving. Cryptography is to protect location privacy by encrypting location points. The method of proxy re-encryption was proposed by Shao et al. [12] for location privacy preserving. Li et al. [13] applied homomorphic cryptography technology to location privacy preserving. Except cryptography, other privacy-preserving technologies are unable to realize the recovery of original information after privacy protection. At the same time, the above methods do not consider the authenticity verification of user's location information.

2.2. Blockchain-Based Location Privacy Preserving. In the current research, Blockchain-based location privacy-preserving schemes all have different degrees of security threats. The decentralized personal data management system was proposed by Li et al. [13], so as to protect user data security. Also, the Blockchain technology was utilized to achieve the automated access control. In VANETs, Li et al. [14] proposed the Blockchain-based trust management algorithm to control the movement behavior of vehicle nodes and achieve the privacy preserving of vehicles. Also, Luo et al. [15] proposed the Blockchain-based location privacy-preserving method by considering the trust mechanism to protect the location privacy of vehicles. The decentralized location privacy-preserving method was proposed by Zhang et al. [16] to protect the location privacy of task and achieve the multi-level location privacy preserving of workers. By making use of the decentralized structure and consensus approach of Blockchain, Zou et al. [17] proposed the two-stage approach realize the nonrepudiation and nontampering of data. Also, it enhanced the sensing quality and protected the data privacy of workers. Most of existing Blockchain-based location privacy-preserving technologies assume that the user who sends the location in the process of location sharing is honest and cannot verify whether the location information after privacy protection is true and credible.

2.3. Encryption Algorithms. Order Preserving Encryption (i.e., OPE) [18–20] was first proposed by Agrawal [18] in 2004. After OPE, the ciphertext retains the original order of the plaintext. Therefore, the size relation of plaintext data can be obtained by comparing the ciphertext directly. For plaintext $x < y$, the OPE ciphertext of x is smaller than the OPE ciphertext of y. For protecting the confidentiality of the data stored in the database, if the traditional encryption method was used, the performance of the data query should be reduced. OPE was a deterministic encryption mechanism, which not only guaranteed the confidentiality of data but also realized the efficient query of data. Initially, OPE was used in databases to perform scoped queries. Then, Boldyreva et al. [19] proposed a security concept based on Pseudorandom Function (PRF), which required the OPE scheme to be as random as possible while preserving the constraints of order. This algorithm was based on the natural relationship between random OPE and hypergeometric probability distribution. And, a kind of order-preserving symmetric encryption was designed by using the black box sampling algorithm of hypergeometric probability distribution. Meanwhile, Boldyreva et al. [19] demonstrated that,

for any $x \in [m]$, the ciphertext $E_{m,n}(x,k)$ of the constructed OPE symmetric encryption mechanism was computationally indistinguishable from the ciphertext $E^*_{m,n}(x,k)$ of the ideal encryption object. Thus, the OPE symmetric encryption mechanism is secure. The OPE in the B-PPLS scheme proposed in this paper all represent the OPE symmetric encryption mechanism in literature [19].

Xiao and Yen [21] proved that when the attacker has m known OPE plaintext and ciphertext combinations, for the OPE ciphertext $E_{m,n}(x,k)$, the attacker is trying to recover the plaintext information x. If it satisfies $h = o(m^e), 0 < e < 1$, and $m^3 \le n$, the probability that the attacker restores plaintext x is a negligible function of security parameter $logm$. These results not only improve the understanding of OPE security but also provide theoretical guidance for the selection of OPE parameters in different application scenarios.

Merkle et al. first proposed the Merkle hash tree [22–24] and designed a data structure to support the verification of retrieval results. The Merkle hash tree is used to verify the integrity of data, and the core idea is to construct the binary tree by using the one-way hash function. The leaf node of the Merkle hash tree is the hash value of the data, and the value of each nonleaf node is the hash value of its two children combined.

Sahai and Waters [25] first mentioned the concept of Attribute-Based Encryption (i.e., ABE) in 2005. In ABE [26–28], the user's identity is described by a series of attributes. When the data Owner encrypts the message, the relevant property access structure is formulated on the property. Only when the attributes owned by the data requester satisfy the attribute access structure set in advance by the encrypter, it can correctly decrypt the ciphertext through the private key.

3. Overview of B-PPLS Scheme

In this section, we introduce the system model, attack model, and security objectives of our proposed B-PPLS.

3.1. System Model. Figure 2 shows the system model of B-PPLS. In B-PPLS, it mainly includes three roles, i.e., Owners, Requesters, and Miners. Owners publish the location data on the Blockchain. In this case, the data recorded on each block of Blockchain is not the transaction in Bitcoin (BTC). It records different Owners' location data arranged in the chronological order. However, the location data is not necessarily continuously recorded on the Blockchain. The Requesters and Owners can share the location data with each other according to the "dual-channel" interaction. If Requester B requests the location data of Owner A at a time, the location data can be requested and verified by the "dual-channel" interaction on and off the chain. The detailed division of each role is summarized as follows.

3.1.1. Owners. The location data can be generated and published by Owners. Miners record the location data on the Blockchain, so as to conduct the further location data

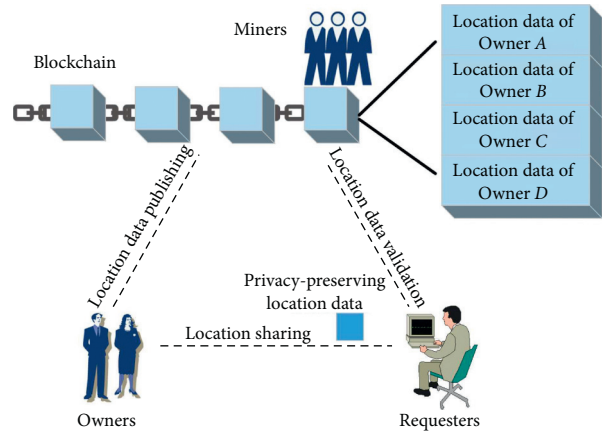

Figure 2: The system model of B-PPLS.

requesting and location data validation for requesters. When the location data is requested at a time, the Owners can implement different levels of location privacy protection according to different identities of requesters. For the multilevel privacy protection, different levels of location privacy protection are represented by different sizes of anonymous regions or original location coordinates. In LBSNs, if the relationship between Owner A and requester B is completely trusted, the precise location data of Owner A can be shared to requester B. However, if the relationship between Owner A and requester B is not completely trusted, the anonymous location region can be shared to requester B in order to protect the personal privacy of Owner A.

3.1.2. Requesters. Requesters send the location data request to the Owners. Different requesters have the different need for the accuracy of location data. For example, the precise location data is needed for positioning service, in order to acquire the real location information of users in LBSNs. However, it is not necessary to provide the precise location data for recommendation service. It can recommend the suitable location-based service for target users according to the location region. Under the condition of "dual-channel" interaction, requesters request the location data of Owners through the interaction off the Blockchain. The strength of privacy protection is determined by the location data Owners. If the Owners provide the location region to the Requesters, the integrity and authenticity of the location region can be verified by the requesters through the interaction on the Blockchain. Also, if the Owners provide the precise location data to the Requesters, the integrity and authenticity of precise location data can be verified.

3.1.3. Miners. Any user in the B-PPLS system can act as a miner. The job of Miners is to collect and check the location information published by the Owners. Miners add new blocks to the Blockchain through the consensus mechanism, e.g., Proof of Work (PoW). That is to say, Miners are responsible for maintaining the steady growth of the Blockchain and ensuring the safety of the B-PPLS system.

B-PPLS can protect the location privacy of users without disclosure in the process of location sharing for LBSNs. It provides better user preference when the shared location may not be precise. Besides, the Blockchain technology is taken into account in B-PPLS, which can well verify the authenticity and integrity of location data.

3.2. Attack Model. In the B-PPLS system, all three roles have potential attack activity. For Owners, the fake locations are shared to the requesters off the Blockchain. Even the Owners deny or falsify the published location record on the Blockchain. For Requesters, the scope of the shared location region is reduced, in order to deduce the precise location of Owners. For Miners, the location data on the Blockchain may be compromised or unauthorizedly accessed. The detailed explanation is as follows.

3.2.1. The Attack on Owners. For the Owners, there are two reasonable assumptions. The one is to assume that the registration information in the Blockchain uploaded by Owners is real. Since each Owner only needs to register the information once before participating in the recording location, this process can be ensured by adding the necessary regulatory means, such as binding with a personal identity. The other one is to assume that the location coordinate information in the Blockchain uploaded by Owners is real. Existing location verification schemes, such as location cryptography secure location protocol [29] and location proof mechanism based on Blockchain [30], can effectively prevent Owners from forging the location coordinate information. Therefore, the above two assumptions are reasonable.

The potential attack behavior of Owners can be divided into two conditions according to different application scenarios. On the one hand, the malicious Owners can send fake location data to the Requesters without Blockchain. On the other hand, malicious Owners can modify the location data and send to the Blockchain.

3.2.2. The Attack on Requesters. Because of multilevel privacy preserving for the B-PPLS scheme, the Requester may obtain the area instead of the exact location of the Owner. At the same time, Requesters with different trust levels can obtain location regions of different sizes. In order to provide more accurate location service, there are two attack behaviors for Requests which are not fully trusted. One is that the Requester attempts to narrow the range of the location area returned by the Owner, that is, to obtain the location area with the low privacy protection level. The other one is that the malicious Requester tries to deduce the exact location coordinates of the Owner.

3.2.3. The Attack on Miners. Because Miners are not involved in the exchange of location information between the Owners and Requesters, they do not have any location information about the Owners. However, the location information of the Owners is recorded in the Blockchain. Due to the characteristics of openness of the Blockchain, malicious Miners may attempt to gain unauthorized access to location information in the Blockchain or cause a leak of location information. That is, without Owners' authorization, malicious Miners may access the personal location information recorded by Owners in the Blockchain.

3.3. Security Objectives. For the traditional location data sharing scheme, as the location data is centrally managed, Owners are not controllable to the location data, and there are risks of malicious disclosure and illegal use. To solve these problems, this paper first proposed a decentralized scheme, that is, the scheme in this paper realized the decentralized management of Owners' location data. Secondly, according to the security threats and the security requirements mentioned in the attack model, the security location sharing scheme should meet the following security attributes.

(1) *Immutability*. The scheme needs to ensure the immutability of Owners on location records. The tampering or falsification of location data will bring many unpredictable security risks.

(2) *Confidentiality*. Attacks can infer the movement trajectories of victims according to the obtained location data. Due to the openness of Blockchain, location sharing using Blockchain will make the location information face the risk of unauthorized access. The scheme should ensure the confidentiality of location data. Specifically, in addition to the requesters authorized by Owners, no other identity can obtain the location information of Owners through the content on the Blockchain.

(3) *Multilevel Privacy-Preserving*. As more and more services use location data, requesters can be users of a variety of identities. However, Owners have different degrees of trust for requesters with different identities. In order to increase the controllability and flexibility of location privacy protection, Owners implement different levels of privacy protection for requesters with different levels of trust. The scheme needs to set up different levels of location information privacy protection.

(4) *Verifiability*. Because of multilevel privacy protection of location information, the fully trusted requesters will obtain the original location coordinates of Owners. Requesters that are not fully trusted will get the privacy-preserving location area of the Owners. In both cases, in order to prevent Owners from deceiving or falsifying the shared location data, requesters need to verify the integrity and authenticity of the obtained location coordinates or location regions.

(5) *Restorability*. Due to the immutability of Blockchain, if other noncryptographic schemes such as k-anonymity and other privacy protection technologies are used, uploading the privacy-preserving location data to the Blockchain will lead to permanent loss of data

quality. The adoption of the cryptography method can meet the needs of completely restoring the location information to the original location after privacy protection.

4. B-PPLS Scheme Designs

In this section, we elaborate the B-PPLS scheme designs. Figure 3 shows the total flowchart of B-PPLS, which can be divided into four stages, i.e., initialization, location record, location sharing, and location verification according to different functions. The B-PPLS scheme ensures that the Owners can only pass the location verification if the real location coordinates and the real location region are shared in the stage of location sharing. The detailed process is described as following.

4.1. Initialization. Algorithm 1 shows the pseudocode of B-PPLS initialization. The main idea of this algorithm is to generate the Merkle hash tree. The Owners specify the geographically rectangular areas to represent the maximum range of activity which can be accessed. The rectangular area R is transformed in Cartesian coordinates, i.e., $R = \{(x, y) | 0 \le x \le X, 0 \le y \le Y\}$. The areas R are iteratively partitioned in the quadtree manner as follows:

$$\left\{ x_i | 1 \le i \le 2^N \wedge x_i = i * \frac{X}{\left(2^N\right)} \right\} \cup \cdot \left\{ y_i | 1 \le i \le 2^N \wedge y_i = i * \frac{Y}{\left(2^N\right)} \right\} \leftarrow \mathrm{Parti}(R, N), \tag{1}$$

where N not only denotes the maximum number of partitions but also denotes the different levels of location privacy protection. Since Algorithm 1 needs to perform OPE operation on all dividing lines, the selection of N should meet the following formula:

$$2^N = o\left(\min(X, Y)^e\right), \quad 0 < e < 1. \tag{2}$$

Owners perform OPE operations one by one along the divider lines perpendicular to the x-axis, i.e., $\mathrm{ciph} \leftarrow \mathrm{OPE}_{X,X'}(k_o^x, x_i)$, $1 \le i \le 2^N$. OPE stands for order-preserving encryption, X and X' are the plaintext space and cryptogram space, respectively, k_o^x is the secret key of Owners, and $X^3 \le X'$ according to the security requirements of OPE. The operation perpendicular to the y-axis is similar to the operation perpendicular to the x-axis, and $k_o^y \ne k_o^x$.

All the splitter plaintext perpendicular $\{x_1, x_2, \ldots, x_{2^N}\}$ to the x-axis is bound one by one with the order-preserving encrypted splitter ciphertext $\{\mathrm{ciph}_1^x, \mathrm{ciph}_2^x, \ldots, \mathrm{ciph}_{2^N}^x\}$ by Owners. It generates $\mathrm{node}_i^x \leftarrow \mathrm{Hash}(i \| x_i \| \mathrm{ciph}_i^x)$, $1 \le i \le 2^N$, by the hash algorithm, and the Merkle hash tree xTree is generated as $x\mathrm{Tree} \leftarrow \mathrm{gen}MT(\mathrm{node}_1^x, \mathrm{node}_2^x, \ldots, \mathrm{node}_{2^N}^x)$. In a similar way, the Merkle hash tree yTree is generated as $y\mathrm{Tree} \leftarrow \mathrm{gen}MT(\mathrm{node}_1^y, \mathrm{node}_2^y, \ldots, \mathrm{node}_{2^N}^y)$.

Finally, the registration record $\mathrm{Reg} \leftarrow \mathrm{id}_o \| x\mathrm{Tree}_{\mathrm{root}} \| x\mathrm{Tree}_{\mathrm{root}} \| \mathrm{signature}_{\mathrm{reg}}$ is generated by signing the root node of the Merkle hash tree xTree and yTree. Thereinto, $\mathrm{signature}_{\mathrm{reg}} \leftarrow \mathrm{ASE}(\mathrm{Pri}_o, \mathrm{Hash}(\mathrm{id}_o \| x\mathrm{Tree}_{\mathrm{root}} \| y\mathrm{Tree}_{\mathrm{root}}))$, ASE is the asymmetric encryption algorithm, and Pri_o is the private key of the Owners. Thus, the operation of the initialization phase is completed.

4.2. Location Record. Each location record of the Owners is generated according to the information of the last uploaded location record. Although the location record of the Owners is not continuously stored in the Blockchain, it can be traced forward based on the address of the previous location record in the current location record. Algorithm 2 shows the pseudocode of the location record generation algorithm.

The main idea of this algorithm is to generate the location record for Owners. Suppose that the Owner generates the ith location coordinate $l_i = (x_i, y_i)$ and uploads the location information to the Blockchain. The order-preserving encryption of l_i is operated by Owners, i.e., $\mathrm{ciph}_i^x \leftarrow \mathrm{OPE}_{X,X'}(k_o^x, x_i)$ and $\mathrm{ciph}_i^y \leftarrow \mathrm{OPE}_{Y,Y'}(k_o^y, y_i)$. Also, the OPE cipher of location l_i is generated as $\mathrm{ciph}_i \leftarrow \mathrm{ciph}_i^x \| \mathrm{ciph}_i^y$. The hash algorithms $\mathrm{OpeHash}_i \leftarrow \mathrm{Hash}(\mathrm{ciph}_i)$ and $\mathrm{LH}_i \leftarrow \mathrm{Hash}(x_i \| y_i)$ are operated for the cipher and plaintext of location $l_i = (x_i, y_i)$, respectively. The symmetric encryption of location l_i plaintext is computed as $\mathrm{SymCiph}_i \leftarrow \mathrm{SE}(k_{\mathrm{sym}}, x_i \| y_i)$. The location information of the Owner at the ith location is denoted as $\mathrm{LI}_i \leftarrow \mathrm{OpeHash}_i \| \mathrm{LH}_i \| \mathrm{SymCiph}_i \| \mathrm{Timestamp}_i$. The location record of the Owner is generated as $\mathrm{LR}_i \leftarrow \mathrm{Pub}_o \| \mathrm{LI}_i \| \mathrm{recId}_{j-1}^o \| \mathrm{signature}_{\mathrm{Loc}}$. Thereinto, $\mathrm{signature}_{\mathrm{Loc}} \leftarrow \mathrm{ASE}(\mathrm{Pri}_o, \mathrm{Hash}(\mathrm{Pub}_o \| \mathrm{LI}_i \| \mathrm{recId}_{j-1}^o))$ is the signature of the location record for the Owner.

Finally, the location record L_i is broadcasted by the Owner and uploaded to the Blockchain by Miners according to the consensus mechanism. Thus, the operation of location record generation is completed.

4.3. Location Sharing. Suppose that the Requester requests the ith location information of the Owner; the location sharing phase can be divided into two categories. One is that the Owner has full confidence in the Requester; then, the Owner returns the exact location coordinates (x_i, y_i). The other one is that the Owner has no full confidence in the Requester; then, the Owner returns the rectangular location area containing location coordinates (x_i, y_i). Algorithm 3 shows the pseudocode of the location sharing algorithm.

When the Owner has full confidence in the Requester, the privacy-preserving level is $n = 0$. It means that the operation of privacy preserving is not performed. Firstly, Owners compute $\mathrm{conRes} \leftarrow \mathrm{SE}(k_{\mathrm{session}}, x_i \| y_i)$ in order to encrypt the exact location. Then, Owners send $\mathrm{Res} \leftarrow \mathrm{conRes} \| \mathrm{ASE}(\mathrm{Pub}_o, k_{\mathrm{session}})$ to Requesters. Finally, the session key k_{session} is obtained by making use of the private key of the Requester, in order to decrypt the location coordinates of the Owner, i.e., $x_i' \| y_i' \leftarrow \mathrm{SD}(k_{\mathrm{session}}, \mathrm{conRes})$.

When the Owner has no full confidence in the Requester, the operation of privacy preserving is necessary for location coordinates (x_i, y_i). Firstly, the Owner computes the privacy-preserving level $n (1 \le n \le N)$ of the Requester and finds the privacy zone boundary $\{x_{\mathrm{id1}}, x_{\mathrm{id2}}, y_{\mathrm{id3}}, y_{\mathrm{id4}}\}$ surrounding

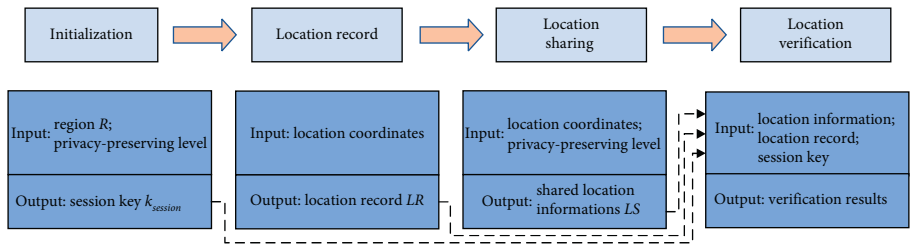

Figure 3: The total flowchart of B-PPLS.

Input: Region $R = \{(x, y) | 0 \le x \le X, 0 \le y \le Y\}$; Maximum level of location partition N; Owner's secret key $k_o = k_o^x \| k_o^y$; Owner's private key Pri_o
Output: Registration record Reg
(1) $\{x_1, x_2, \ldots, x_{2^N}\} \cup \{y_1, y_2, \ldots, y_{2^N}\} \leftarrow \text{Parti}(R, N)$
(2) **For** $i \longrightarrow 1, 2, \ldots, 2^N$ **do**
(3) $\text{ciph}_i^x = E_{X,X'}(k_o^x, x_i)$;
(4) $\text{ciph}_i^y = E_{Y,Y'}(k_o^y, y_i)$;
(5) $\text{node}_i^x = \text{Hash}(i \| x_i \| \text{ciph}_i^x)$;
(6) $\text{node}_i^y = \text{Hash}(i \| y_i \| \text{ciph}_i^y)$;
(7) **End for**
(8) $x\text{Tree} \leftarrow \text{gen}MT(\text{node}_1^x, \text{node}_2^x, \ldots, \text{node}_{2^N}^x)$;
(9) $y\text{Tree} \leftarrow \text{gen}MT(\text{node}_1^y, \text{node}_2^y, \ldots, \text{node}_{2^N}^y)$;
(10) $R \leftarrow \text{id}_o \| x\text{Tree}_{\text{root}} \| y\text{Tree}_{\text{root}} \| \text{signature}_{\text{reg}}$
(11) **Return** Reg

Algorithm 1: B-PPLS initialization.

Input: Owner's ith location (x_i, y_i); Owner's secret keys $k_o = k_o^x \| k_o^y, k_{\text{sym}}$
Output: Location record LR
(1) **Owners execute**:
(2) $\text{ciph}_i^x = E_{X,X'}(k_o^x, x_i)$
(3) $\text{ciph}_i^y = E_{Y,Y'}(k_o^y, y_i)$;
(4) $\text{ciph}_i \leftarrow \text{ciph}_i^x \| \text{ciph}_i^y$;
(5) $\text{OpeHash}_i \leftarrow \text{Hash}(\text{ciph}_i)$;
(6) $\text{LH}_i \leftarrow \text{Hash}(x_i \| y_i)$;
(7) $\text{SymCiph}_i \leftarrow \text{Enc}(k_{\text{sym}}, x_i \| y_i)$;
(8) $\text{LI}_i \leftarrow \text{OpeHash}_i \| \text{LH}_i \| \text{SymCiph}_i \| \text{timestamp}_i$;
(9) $\text{LR}_i \leftarrow \text{Pub}_o \| \text{LI}_i \| \text{recId}_{i-1}^o \| \text{signature}_o$;
(10) **Return** LR

Algorithm 2: Location record generation.

location l_i under the privacy protection level n in region R. The zone divider information can be generated as $\text{borInfo} \leftarrow \text{borInfo}_{\text{id1}} \| \text{borInfo}_{\text{id2}} \| \text{borInfo}_{\text{id3}} \| \text{borInfo}_{\text{id4}}$; thereinto,

$$\begin{aligned} \text{borInfo}_{\text{id1}} &\leftarrow \text{id}_1 \| x_{\text{id1}} \| \text{ciph}_{\text{id1}}^x, \\ \text{borInfo}_{\text{id2}} &\leftarrow \text{id}_2 \| x_{\text{id2}} \| \text{ciph}_{\text{id2}}^x, \\ \text{borInfo}_{\text{id3}} &\leftarrow \text{id}_3 \| y_{\text{id3}} \| \text{ciph}_{\text{id3}}^y, \\ \text{borInfo}_{\text{id4}} &\leftarrow \text{id}_4 \| x_{\text{id4}} \| \text{ciph}_{\text{id4}}^x. \end{aligned} \tag{3}$$

In order to realize the integrity verification of the location frame for the Requester during the stage of location verification, the Owner should return the Merkle hash tree $x\text{Tree}$ and $y\text{Tree}$, i.e., $\text{nodes}^x \leftarrow \{\text{node}_{x1}^x, \text{node}_{x2}^x, \ldots\}$ and $\text{nodes}^y \leftarrow \{\text{nodes}_{y1}^y, \text{nodes}_{y2}^y, \ldots\}$. The Owner computes $\text{fuzRes} \leftarrow \text{Enc}(k_{\text{session}}, \text{ciph}_i \| \text{borInfo} \| \text{nodes}^x \| \text{nodes}^y)$ and returns $\text{Res} \leftarrow \text{fuzRes} \| \text{ASE}(\text{Pub}_r, k_{\text{session}})$ to the Requester through the underchain channel. Finally, the location information of the Owner is decrypted by making use of the session key k_{session} generated by the private key Pri_r of the Requester. Because the border plaintext information $\{x'_{\text{id1}}, x'_{\text{id2}}, y'_{\text{id3}}, y'_{\text{id4}}\}$ is contained in $\text{borInfo}'$, the privacy-preserving location area is obtained by the Requester. The rest part of fuzRes is used to the location verification stage for the Requester. Thus, the operation of location sharing for the Owner and Requester is completed.

Input: Privacy-preserving level n; Owner's public key Pub_o; Owner's location $l_i = (x_i, y_i)$
Output: Shared location information LS
(1) **Owners execute**:
(2) **If** $n = 0$ **then**
(3) $\text{conRes} \leftarrow \text{SE}(k_{\text{session}}, x_i \| y_i)$;
(4) $\text{Res} \leftarrow \text{conRes} \| \text{ASE}(\text{Pub}_o, k_{\text{session}})$;
(5) **Else If** $1 \le n \le N$ **then**
(6) find the border $\{x_{\text{id1}}, x_{\text{id2}}, y_{\text{id3}}, y_{\text{id4}}\}$ in level n;
(7) $\text{borInfo}_{\text{id}_1} \leftarrow \text{id}_1 \| x_{\text{id1}} \| \text{ciph}^x_{\text{id1}}$;
(8) $\text{borInfo}_{\text{id}_2} \leftarrow \text{id}_2 \| x_{\text{id2}} \| \text{ciph}^x_{\text{id2}}$;
(9) $\text{borInfo}_{\text{id}_3} \leftarrow \text{id}_3 \| y_{\text{id3}} \| \text{ciph}^y_{\text{id3}}$;
(10) $\text{borInfo}_{\text{id}_4} \leftarrow \text{id}_4 \| y_{\text{id4}} \| \text{ciph}^y_{\text{id4}}$;
(11) $\text{borInfo} \leftarrow \text{borInfo}_{\text{id}_1} \| \text{borInfo}_{\text{id}_2} \| \text{borInfo}_{\text{id}_3} \| \text{borInfo}_{\text{id}_4}$;
(12) $\text{node}^x \leftarrow \{\text{node}^x_{x1}, \text{node}^x_{x2}, \ldots\}$;
(13) $\text{node}^y \leftarrow \{\text{node}^y_{y1}, \text{node}^y_{y2}, \ldots\}$;
(14) $\text{fuzRes} \leftarrow \text{Enc}(k_{\text{session}}, \text{ciph}_i \| \text{borInfo} \| \text{nodes}^x \| \text{nodes}^y)$;
(15) $\text{Res} \leftarrow \text{fuzRes} \| \text{ASE}(\text{Pub}_r, k_{\text{session}})$;
(16) **End If**
(17) **Requesters execute**:
(18) **If** $n = 0$ **then**
(19) $\| x'_i y'_i \leftarrow \text{SD}(k_{\text{session}}, \text{conRes})$;
(20) $\text{LS} \leftarrow x'_i \| y'_i$;
(21) **Else If** $1 \le n \le N$ **then**
(22) $\text{ciph}'_i \| \text{borInfo}' \| \text{nodes}'^x \| \text{nodes}'^y \leftarrow \text{SD}(k_{\text{session}}, \text{fuzRes})$;
(23) $\text{LS} \leftarrow \text{ciph}'_i \| \text{borInf}' \| \text{nodes}'^x \| \text{nodes}'^y$;
(24) **End If**
(25) **Return** LS

Algorithm 3: Location sharing.

4.4. Location Verification. Suppose that the Requester verifies the ith location information of the Owner during the location sharing; the location verification phase can be divided into two categories corresponding to the location sharing phase. Algorithm 4 shows the pseudocode of the location verification algorithm.

When the Owner has full confidence in the Requester, the precise location coordinates $l'_i = (x'_i, y'_i)$ of the Owner can be acquired after conRes is decoded by the Requester. Then, the Requester retrieves the location record LR_i on the Blockchain generated during the location record phase. If $\text{Hash}(x'_i \| y'_i) = \text{LR}_i \cdot \text{LI}_i \cdot \text{LH}_i$, it shows that $x'_i = x_i$ and $y'_i = y_i$; the Requester gets the correct location coordinate information of the Owner. Otherwise, the validation fails.

When the Owner has no full confidence in the Requester, $\text{ciph}'_i \| \text{borInfo}' \| \text{nodes}'^x \| \text{nodes}'^y \leftarrow \text{SD}(k_{\text{session}}, \text{fuzRes})$ is acquired after fuzRes is decoded by the Requester. nodes'^x and nodes'^y are the required nodes on the root node authentication path for recovering xTree and yTree. Firstly, the Requester verifies the integrity of the received region boundary information borInfo′. If $\text{genMT}(\text{Hash}(\text{borInfo}'_{\text{id1}}), \text{Hash}(\text{borInfo}'_{\text{id2}}), \text{nodes}'^x) = x\text{Tree}'_{\text{root}}$ and $\text{genMT}(\text{Hash}(\text{borInfo}'_{\text{id1}}), \text{Hash}(\text{borInfo}'_{\text{id2}}), \text{nodes}'^y) = y\text{Tree}'_{\text{root}}$, the regional integrity verification is successful; it shows that the correct region information $\text{borInfo}' = \text{borInfo}$ is returned by the Owner. Otherwise, the region verification fails. Then, the Requester verifies the authenticity of the received region boundary information borInfo′. If $\text{Hash}(\text{ciph}'_i) = \text{LR}_i \cdot \text{LI}_i \cdot \text{OpeHash}_i$, $(\text{borInfo}'_{\text{id1}} \cdot \text{ciph}^x_{\text{id1}}) \le (\text{ciph}'_i \cdot \text{ciph}^x_i) \le (\text{borInfo}'_{\text{id}\,2} \cdot \text{ciph}^x_{\text{id}\,2})$, and $(\text{borInfo}'_{\text{id}\,3} \cdot \text{ciph}^y_{\text{id}\,3}) \le (\text{ciph}'_i \cdot \text{ciph}^y_i) \le (\text{borInfo}'_{\text{id}\,4} \cdot \text{ciph}^y_{\text{id}4})$, it shows that $x_{\text{id1}} \le x_i \le x_{\text{id2}}$ and $y_{\text{id}\,3} \le y_i \le y_{\text{id}\,4}$; location l_i is in the region surrounded by $\{x_{\text{id1}}, x_{\text{id2}}, y_{\text{id3}}, y_{\text{id4}}\}$ that the Requester receives. At this point, the region verification is successful. Otherwise, the region verification fails. Finally, the operation of location verification for the Owner and Requester is completed.

The computational complexity of the above four algorithms are O (n), O (1), O (1), and O (1), respectively. The computation overhead of Algorithm 1 increases exponentially with the increase of N. Although large plaintext space brings high computation overhead, it can be realized offline in the initialization phase. Furthermore, Algorithm 1 is executed only one time during the total process. The computation overhead of Algorithm 2 increases with the increase of plaintext space, which is almost unaffected by the value of N. Also, it has small computation overhead in the location record phase. The computation overhead of Algorithm 3 is almost unaffected by the two factors of N and plaintext space. It takes less time at the location sharing phase for Owners. The computation overhead of Algorithm 4 is approximately linear with the size of N. It is not affected by the size of the plaintext space. Since only hash operation is involved in the location verification phase, the computation overhead is very small.

5. Security Analysis

In this section, the security analysis is given corresponding to the five security attributes in the security objectives of Section 3, in order to prove the security of the B-PPLS scheme.

Input: Location sharing information LS; Location record in the Blockchain LR; session key $k_{session}$
Output: Boolean variable LV
(1) **Initialize** LV←False
(2) **If** $n = 0$ **then**
(3) $x_i' \| y_i' \leftarrow$LS;
(4) **If** Hash $(x_i' \| y_i')$LR$_i \cdot$ LI$_i \cdot$ LH$_i$ **then**
(5) LV←True;
(6) **End If**
(7) **Else If** $1 \le n \le N$ **then**
(8) ciph$_i' \| $borInfo$' \| $nodes$'^x \| $nodes$'^y \leftarrow$LS;
(9) borInfo$_{id1}' \|$borInfo$_{id2}' \|$borInfo$_{id3}' \|$borInfo$_{id4}' \leftarrow$borInfo$'$;
(10) xTree$_{root}' \leftarrow$genMT (Hash (borInfo$_{id1}'$), Hash (borInfo$_{id\,2}'$), nodes$'^x$);
(11) yTree$_{root}' \leftarrow$genMT (Hash (borInfo$_{id1}'$), Hash (borInfo$_{id\,2}'$), nodes$'^y$);
(12) **If** xTree$_{root}' = x$Tree$_{root}$ and yTree$_{root}' = y$Tree$_{root}$ **then**
(13) **If** Hash (ciph$_i'$) = LR$_i \cdot$ LI$_i \cdot$ OpeHash$_i$ and borInfo$_{id1}' \cdot$ ciph$_{id1}^x <$ ciph$_i' \cdot$ ciph$_i^x <$ borInfo$_{id2}' \cdot$ ciph$_{id2}^x$ and borInfo$_{id3}' \cdot$ ciph$_{id3}^y <$ ciph$_i' \cdot$ ciph$_i^y <$ borInfo$_{id\,4}' \cdot$ ciph$_{id\,4}^y$ **then**
(14) LV←True;
(15) **End If**
(16) **End If**
(17) **End If**
(18) **Return** LV

Algorithm 4: Location verification.

5.1. Immutability. According to the principle of Blockchain, all subsequent blocks block$_i$ $(1 < i)$ are constructed from the hash value Hash(block$_{i-1}$) of the previous block, with the exception of the creation block. According to the difficulty value set by the system, the correct random number is found first so that the hash value of the data in the block header is less than or equal to the target hash value. Miners who have been verified by the whole system get the accounting right and are connected after block$_{i-1}$. The attacker who tampers with a record in the block must recalculate the SHA256 puzzle for that block and all subsequent blocks. The computational force must make the forged bifurcated chain become longer than the main chain. If the Owner modifies the location record in block$_{i-1}$, the proof of work must be redone for all blocks block$_j$ $(j > i)$ linked after it. A new fork is created, in order to make the chain length of the fork larger than that of the main chain of original Blockchain. Since the computing power mastered by the Owner is less than 51% of the whole system, the cost required to master 51% of the computing power in the actual system far exceeds the benefits obtained after a successful attack; the probability of the Owner producing a longer fork than the original Blockchain is negligible. Thus, the probability ϵ_1 of location repudiation is negligible, while Owners illegally modify the location information already uploaded to the Blockchain.

5.2. Confidentiality. There are two types of records in the Blockchain, namely, the location record and the registration record. For the location records LR$_i$ = Pub$_o \| LI_i \|$ recId$_{j-1}^o \|$signature$_{Loc}$, thereinto LI$_i$ = OpeHash$_i \|$LH$_i \|$ SymCiph$_i \|$Timestamp$_i$. In LR$_i$, the ones that contain the location information of Owners are OpeHash$_i$, LH$_i$, and SymCiph$_i$. Firstly, OpeHash$_i$ = Hash(ciph$_i$) and LH$_i$ = Hash$(x_i | y_i)$; if Miners recover ciph$_i$ or $x_i \| y_i$ with the nonnegligible probability ε_h, the unidirectivity of the unidirectional hash function is contradicted. Therefore, the probability ε_h is negligible. Second, SymCiph$_i$ = SE$(k_{sym}, x_i \| y_i)$; if Miners, with the nonnegligible probability ε_s, can successfully decrypt the symmetric encryption plaintext without having the private key k_{sym} of the Owner, the confidentiality of the symmetric encryption would be contradicted. Therefore, the probability $\varepsilon_l = \varepsilon_h + \varepsilon_s$ of Miners making unauthorized access based on location information LR$_i$ in the Blockchain is negligible.

For the registration records Reg = id$_o \| x$Tree$_{root} \|$ yTree$_{root} \|$signature$_{reg}$, thereinto xTree$_{root}$ and yTree$_{root}$ are both root nodes of the Merkle hash tree. The root nodes are generated by the one-way hash operation performed iteratively by the leaves of the Merkle hash tree. If Miners recover the root of the Merkle hash tree with nonnegligible probability ε_r, the unidirectivity of the hash function is contradicted. Therefore, the probability ε_r of Miners inferring the plaintext of location information from the root node of the Merkle hash tree is negligible.

In conclusion, the probability $\varepsilon_2 = \varepsilon_l + \varepsilon_r$ that the attacker can access the location information of Owners according to the records in the Blockchain is negligible under the condition of no authorization.

5.3. Multilevel Privacy Preserving. In the process of location sharing, the size of the location area obtained by the Requester that is not fully trusted is determined by the level of privacy protection. The privacy protection level corresponding to the Requester is set by the Owner according to the trust level. According to Algorithm 1, the region R can be divided into the grid of different sizes at N levels. For the Requester, the Owner finds the location region partition line containing the original location point under the privacy-

preserving level and returns the plaintext ciphertext combination of the four partition lines of the region to the Requester. The Requester determines the location of the region according to the plaintext x_{id1}, x_{id2}, y_{id3}, and y_{id4} of the four dividing lines and verifies the integrity and authenticity of the dividing lines according to the ciphertext $ciph^{x}_{id1}$, $ciph^{x}_{id2}$, $ciph^{y}_{id3}$, and $ciph^{y}_{id4}$ of the four dividing lines.

If the Requester that is not fully trusted requests multiple locations from the Owners, each location region returned by the Owner consists of a combination of four split-line plaintext and order-preserving encrypted ciphertext, i.e., $borInfo_{id_1} = id_1 \| x_{id1} \| ciph^{x}_{id1}$. The Requester can obtain the plaintext and ciphertext combination of multiple dividing lines, including $\{borInfo_{i\,d\,x1}, borInfo_{i\,d\,x2}, \ldots\}$ and $\{borInfo_{i\,d\,y1}, borInfo_{i\,d\,y2}, \ldots\}$. For the B-PPLS scheme, since the Requester is in the privacy-preserving level N, all the position areas generated by the divider line do not cross or overlap under the same privacy protection level. Therefore, it is completely unable to reduce the range of the received region by crossing and overlapping the divided regions for Requesters. There are two kinds of splitters in order-preserving encryption, $k^{x}_{o} \neq k^{y}_{o}$, $X \neq Y$, $X' \neq Y'$; if the probability of the order-preserving cryptography association between $borInfo_x$ and $borInfo_y$ is nonnegligible, it contradicts the security definition of OPE. Therefore, the probability ε_m of the attacker reducing the accepted region size through the association between $borInfo_x$ and $borInfo_y$'s OPE ciphertext is negligible.

In a worst-case scenario, the Requester itself, or through collusion, knows all the split-line plaintext and ciphertext combinations of the Owner. The Requester has the same number of dividing lines perpendicular to the x-axis and the y-axis, all of which are 2^N. In this case, we are just talking about the line that is perpendicular to the x-axis, and the same is true for the line that is perpendicular to the y-axis. According to reference [21], for any $x \in [m]$, its ciphertext $E_{m,n}(x,k)$ is computationally indistinguishable from $E^{*}_{m,n}(x,f)$ in the ideal OPE object. Thus, the OPE used in the B-PPLS scheme is safe. In the case of OPE security, for attackers with h combinations of plaintext and ciphertext, if $h = o(m^e)$, $(0 < e < 1)$, $m^3 \leq n$. Then, the ciphertext $E_{m,n}(x,k)$ is given to the attacker; the probability that the attacker can completely recover the plaintext is negligible. Assume that the attacker has all the plaintext and ciphertext combinations of the dividing lines in the x-axis direction of the Owner after multiple location requests; the number of which is 2^N. Because $X^3 \leq X'$, thereinto X is the plaintext space of order-preserving encryption, X' is the ciphertext space of order-preserving encryption, and $2^N = o(\min(X,Y)^e)$; it meets the security requirements. Suppose that the Requester knows all the plaintext and ciphertext combinations that are perpendicular to the x-axis. If the position coordinates l_i and abscess x_i of the Owner are decrypted with a nonnegligible probability ε_x, it contradicts the definition of OPE security analysis. The same with the line that is perpendicular to the y-axis. Therefore, the probability $\varepsilon_n = \varepsilon_x * \varepsilon_y$ that the Requester can infer the exact position coordinates of the Owner is negligible.

In conclusion, the probability $\varepsilon_3 = \varepsilon_m + \varepsilon_n$ that the Requester will violate the multilevel privacy preserving of the Owner by reducing the scope of the returned region or deducing the exact location coordinates of the Owner is negligible.

5.4. Verifiability. When the Requester is fully trusted by the Owner, the probability that the false location information returned by the Owner to the Requester can be verified by the Requester is negligible. That is to say, $conRes' \leftarrow SE(k_{session}, x'_i \| y'_i)$, $x_i \neq x'_i$ and $y_i \neq y'_i$; the probability of Requester validation is negligible. In the stage of location verification, it determines that whether $Hash(x'_i \| y'_i)$ is equal to the location record $LR_i \cdot LI_i \cdot LH_i$ that the Owner has uploaded on the Blockchain. If the Owner finds $x_i \neq x'_i$ or $y_i \neq y'_i$ with nonnegligible probability ϵ_{v1} and makes $Hash(x'_i \| y'_i) = LR_i \cdot LI_i \cdot LH_i$, it contradicts the collision resistance of the one-way hash function. Thus, the probability ϵ_{v1} is negligible.

When the Requester is not fully trusted by the Owner, the probability that the false privacy-preserving location information returned by the Owner to the Requester can be verified by the Requester is negligible. That is to say, if the Owner returns $fuzRes' = Enc(k_{session}, ciph'_i \| borInfo' \| nodes'^{x} \| nodes'^{y})$ to the Requester and $ciph'_i \neq ciph_i$, $borInfo' \neq borInfo$, $nodes'^{x} \neq nodes^{x}$, and $nodes'^{y} \neq nodes^{y}$, the probability of the returned location passing validation by the Requester is negligible. In the stage of location verification, the Requester verifies the integrity of $ciph'_i$ and determines whether $Hash(ciph'_i)$ is equal to the location record $LR_i \cdot LI_i \cdot LH_i$ that the Owner has uploaded on the Blockchain. If the probability ϵ_{v2} that the malicious Owner finds $ciph'_i \neq ciph_i$ and satisfies $Hash(ciph'_i) = LR_i \cdot LI_i \cdot LH_i$ is not negligible, it contradicts the collision resistance of the one-way hash function. Thus, the probability ε_{v2} is negligible.

In conclusion, the probability $\varepsilon_4 = \varepsilon_{v1} + \varepsilon_{v2}$ that the attacker Owner returns the wrong location information to the Requester and implements the location deception is negligible.

5.5. Restorability. For the ith location record $LR_i = Pub_o \| LI_i \| recId^{o}_{i-1} \| signature_o$ of the Owner in the Blockchain, the location information $LI_i = OpeHash_i \| LH_i \| SymCiph_i \| timestamp_i$, so the ones in LR_i that contain the location information of the Owner are $OpeHash_i$, LH_i, and $SymCiph_i$. Thereinto, $SymCiph_i = SE(k_{sym}, x_i \| y_i)$, and key k_{sym} is kept secret by the Owner. That is to say, the location record of the Blockchain contains the symmetric encrypted ciphertext of the coordinate $(x_i \| y_i)$ of the specific location; only the Owner can decrypt the ciphertext $SymCiph_i$ to achieve recoverability of the original location.

6. Performance Evaluation

In this section, the performance of the proposed B-PPLS scheme is evaluated according to computation overhead. The experimental results can be divided into four parts, i.e., initialization, location record, location sharing, and location verification. Furthermore, the B-PPLS scheme is compared with other related schemes.

6.1. Datasets and Experimental Setup

6.1.1. Datasets. We use the real datasets to verify the validity and efficiency of our methods. GeoLife datasets [35] have recorded the GPS trajectory of 182 users with 18670 trajectories in five years (from 2008/10 to 2012/8), which not only includes the daily activity (e.g., going home and working) but also includes the recreational activity (e.g., shopping, traveling, eating, and sports). Most of the data in GeoLife datasets lies in Beijing, and few of them in Europe or USA.

6.1.2. Experimental Setup. All experiments are run on a computer with Intel i7-3770 3.40 GHz CPU and 8 GB RAM, 64 bit Windows 10 OS. The location data analysis and privacy-preserving algorithms of four stages are conducted on MATLAB 2014b. Because the trajectory data has been sampled by Geolife, the performance of initialization, location record, location sharing, and location verification in the B-PPLS scheme can be evaluated according to the Geolife datasets. The parameter setup of four stages is same. The hash function is SHA-256. The asymmetric cryptographic algorithm is RSA-1024. The OPE plaintext spaces are $0 - 2^{10}$, i.e., OPE (10), $0 - 2^{20}$, i.e., OPE (20), and $0 - 2^{30}$, i.e., OPE (30). According to the security requirements of OPE, the ciphertext space is set to the third power of the plaintext space.

6.2. Experimental Results on Computation Overhead

6.2.1. Initialization. In the stage of initialization, the region R is first iteratively partitioned N times by the Owner according to the quadtree. It represents that the initialization can achieve N different degrees of privacy preserving. Then, all the separators are performed with the OPE operation. Finally, two Merkle hash trees are generated by dividing lines perpendicular to the x-axis and perpendicular to the y-axis as leaf nodes.

Figure 4(a) shows that the overhead during the initial phase increases exponentially with the increase of N, in the case that the size of the plaintext space is constant. It exceeds 2500 ms, while $N = 8$ and plaintext space is OPE (30). The reason is that the number of separators is 2^N, so 2^N OPE encryption operations are required during the initial phase. Because the overhead of calculating Merkle hash trees is lower than that of the OPE algorithm, the OPE algorithm occupies a large proportion during the initial stage of B-PPLS. Meanwhile, at the same privacy-preserving strength N, the different size of plaintext space for the OPE algorithm causes the difference of overhead. In this stage, the larger plaintext space can cause the slightly higher overhead and make OPE encryption operations less efficient.

Therefore, the computation overhead of B-PPLS in the initialization stage increases exponentially with the increase of N. The larger OPE plaintext space results in higher computation overhead. Although this phase is time-consuming, it can be implemented offline and only needs to be done once throughout the course of the solution.

6.2.2. Location Record. As shown in Figure 4(b), as the privacy protection level N increases, the computation overhead does not change significantly. That is to say, the privacy protection level N does not affect the computation overhead of Owners in the stage of location record. However, the increase of the OPE plaintext space will lead to the increase of computation overhead. When the plaintext space is OPE (10), the computation overhead in this stage is stable at around 24 ms. When the plaintext space is OPE (20), the computation overhead in this stage is stable at around 31 ms. When the plaintext space is OPE (10), the computation overhead in this stage is stable at around 37 ms. The reason is that the efficiency of the OPE algorithm can mainly affect the computation overhead at this stage, and the increase of the plaintext space will lead to the increase of the computation overhead.

Therefore, the overhead of B-PPLS in the location record stage increases with the increase of OPE plaintext space. However, it is almost not affected by the privacy-preserving level N, and the computation overhead of this stage is small overall.

6.2.3. Location Sharing. In the stage of location sharing, the Requesters can be divided into full trust and incomplete trust according to the degree of trust by Owners. When the Requester is not fully trusted, the Owner selects the partition line surrounding its own location according to different privacy-preserving levels for the Requester. The plaintext of the partition line is returned to the Requester after combining it with OPE ciphertext. In this case, the computation overhead of Owners varies with privacy-preserving level N in different plaintext spaces, i.e., OPE (10), OPE (20), and OPE (30). When the Requester is fully trusted, $N = 0$ indicates that the Owner has no privacy preserving for the Requester, and the Owner will return its exact location information to the Requester.

As shown in Figure 4(c), regardless of whether the Requester is fully trusted or not, the broken lines of computation overhead under different parameters almost coincide. It is stable at about 12 ms and do not change with the increase of privacy-preserving level N. The reason is that the main operation performed at the stage of location sharing is symmetric encryption. The OPE ciphertext of the dividing line involved in this stage is generated during the stage of initialization, and the OPE ciphertext of the location information is generated in the stage of location record. If the operation of privacy preserving is conducted, the input length of symmetric encryption is affected by the size of the plaintext space and N. However, in this stage, the input length of symmetric encryption has little change under different parameters, which has almost no impact on the execution efficiency of symmetric encryption. All the parameters mentioned above have little influence on the computation overhead of this stage.

Therefore, whether the Requester is fully trusted or not, the computation overhead of the B-PPLS location sharing phase is almost unaffected by privacy-preserving level N and the OPE plaintext space, and the Owner takes less time in this stage.

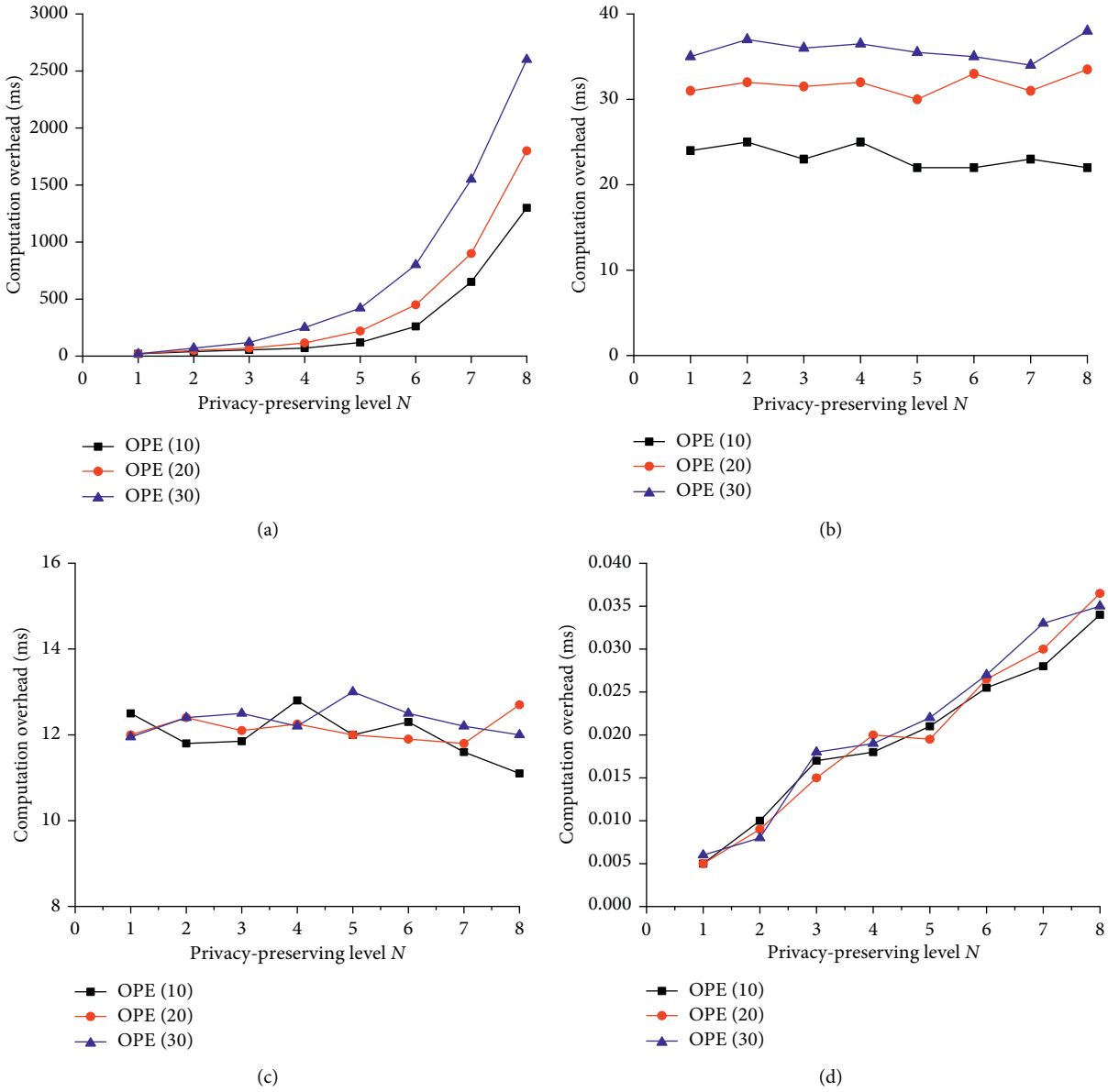

Figure 4: Computation overhead of four stages. (a) Initialization. (b) Location record. (c) Location sharing. (d) Location verification.

6.2.4. Location Verification. In the stage of location verification, the Requesters can also be divided into full trust and incomplete trust according to the degree of trust by Owners. When the Requester is not fully trusted, it verifies the integrity and authenticity of the received regional partition information through the location record LR_i and the registration record Reg_i on the Blockchain. When the Requester is fully trusted, i.e., $N = 0$, it verifies the integrity of the location information only through the location record LR_i of the Owners on the Blockchain.

As shown in Figure 4(d), the computation overhead of three curves corresponding to the different OPE plaintext spaces almost coincides when the Requester is not fully trusted. The reason is that the Requester only compares the results of OPE ciphertext and does not involve the operation of OPE encryption and decryption in the stage of location verification. Thus, the size of the plaintext space has no effect on this stage. The computation overhead increases linearly with the increase of privacy-preserving level N. The reason is that N is equal to the height of the Merkle hash tree in the registered record Reg_i. The height of the Merkle tree determines the length of the authentication path of the root node, that is, the number of hash operations.

Therefore, when the Requester is not fully trusted, the computation overhead of the B-PPLS location verification phase is approximately linear with the size of the privacy-preserving level N and is not affected by the size of the OPE plaintext space. When the Requester is fully trusted, the computational overhead of the Requester is less than that of the Requester that is not fully trusted and is not affected by N. Since only hash operation is involved in the location verification phase, the computational overhead in this stage is very small.

Table 1: Comparison with related schemes.

Schemes	Architectures	Privacy-preserving technologies	Privacy preserving and data utility	Computation overhead
B-PPLS	Decentralized	Blockchain and OPE technology	Good privacy preserving and high data utility	Low
[31]	Centralized	Spatial obfuscation technology	Relying on third parties while getting the lower data utility	High
[32]	Decentralized	Encryption-based technology	Privacy preserving and data utility cannot be well balanced	Medium
[33]	Multiserver	User collaborative-based technology	The better the privacy preserving, the lower the data utility	Medium
[34]	Decentralized	Blockchain technology	Good privacy preserving, but low data utility	Medium
[11]	Decentralized	Differential privacy-based technology	Good privacy preserving and data utility	Medium

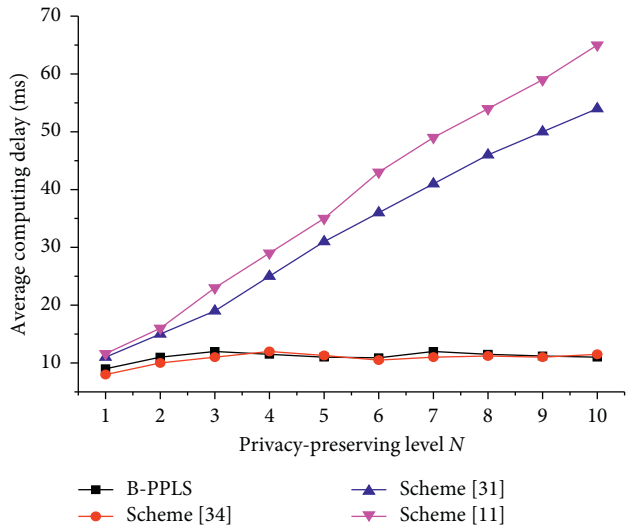

Figure 5: Comparison on average computing delay.

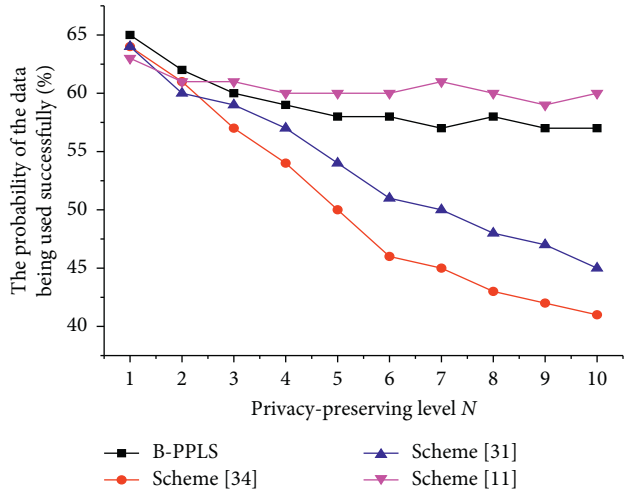

Figure 6: Comparison on data utility.

6.3. Scheme Comparison. The comparison of our proposed B-PPLS scheme with other location privacy-preserving schemes is shown in Table 1. Our proposed B-PPLS scheme makes use of Blockchain and OPE technology to get good privacy preserving while also getting high data utility. Furthermore, the computation overhead of the proposed B-PPLS scheme is low. The scheme [31] relies on the third parties which cause the lower privacy preserving and data utility. The computation overhead of the scheme [31] is also high. Encryption-based technology is used by the scheme [32] in order to protect location privacy. However, the data utility is not guaranteed. The scheme [34] does not use the OPE technology, which leads to lower data utility. The differential privacy-based location privacy-preserving scheme [11] can provide the good privacy preserving and data utility, but the computation overhead is a little high.

Figure 5 shows the comparison on average computing delay. We can see that B-PPLS and scheme [34] have lower average computing delay with privacy-privacy level increase. Due to the characteristics of Blockchain technology, users' location data can be protected effectively. However, the location privacy preserving of the scheme in [31] and [11] is achieved by location obfuscation, which causes the average computing delay increase with the privacy-privacy level increase. As shown in Figure 6, we can see that B-PPLS and scheme [11] have the better data utility than the other two schemes. The reason is that the OPE algorithm in B-PPLS can ensure that the order of the location is unchanged after encryption. Furthermore, because the differential privacy model can well ensure the consistency of the output results, the scheme [11] has the best data utility in four schemes. However, the computation overhead of the scheme [11] is higher than our proposed B-PPLS.

7. Conclusion

In this paper, we have proposed and implemented a Blockchain-enabled privacy-preserving location sharing (B-PPLS) scheme. To be specific, B-PPLS includes Owners, Requesters, and Miners. Owners share the location data to Requesters on the Blockchain. The data recorded on each block records different Owners' location data arranged in the chronological order. Furthermore, four stages (i.e., initialization, location record, location sharing, and location verification) are designed to ensure the security of location sharing. Also, the algorithms corresponding to the stages are proposed, and the security analysis is given for the security objectives. Several simulations are carried out to evaluate the performance of our system. Analysis and evaluation show that our proposed scheme is effective and feasible for the sharing of location data. Further studies are still needed in

the future. For example, how to mine the personalized features of Owners and Requesters in B-PPLS is an important problem to be further studied.

Data Availability

The data used to support the findings of the study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (NSFC) under Grant no. 61902361 and in part by the Henan Provincial Science and Technology Department under Grant nos. 212102210095 and 202102210176 and Key Scientific Research Project of Colleges and Universities in Henan Province under Grant no. 20A120011.

References

[1] L. Zhu, C. Xu, J. Guan, and H. Zhang, "Sem-ppa: a semantical pattern and preference-aware service mining method for personalized point of interest recommendation," *Journal of Network and Computer Applications*, vol. 82, pp. 35–46, 2017.

[2] L. Zhu, L. Yu, Z. Cai, and J. Zhang, "Toward pattern and preference-aware travel route recommendation over location-based social networks," *Journal of Information Science and Engineering*, vol. 35, no. 5, pp. 959–975, 2019.

[3] K. Gai, J. Guo, L. Zhu, and S. Yu, "Blockchain meets cloud computing: a survey," *IEEE Communications Surveys Tutorials*, vol. 22, no. 3, pp. 2009–2030, 2020.

[4] L. Zhu, H. Xie, Y. Liu, J. Guan, Y. Liu, and Y. Xiong, "Ptpp: preference-aware trajectory privacy-preserving over location-based social networks," *Journal of Information Science and Engineering*, vol. 34, no. 4, pp. 803–820, 2018.

[5] X. Liu, K. Liu, L. Guo, X. Li, and Y. Fang, "A game-theoretic approach for achieving k-anonymity in location based services," in *Proceedings IEEE INFOCOM*, pp. 2985–2993, Turin, Italy, April 2013.

[6] F. Fei, S. Li, H. Dai, C. HuC., W. Dou, and Q. Ni, "A k-anonymity based schema for location privacy preservation," *IEEE Transactions on Sustainable Computing*, vol. 4, no. 2, pp. 156–167, 2019.

[7] A. Pingley, W. Yu, N. Zhang, X. Fu, and W. Zhao, "Cap: a context-aware privacy protection system for location-based services," in *Proceedings of the 2009 29th IEEE International Conference on Distributed Computing Systems*, pp. 49–57, Quebec, Canada, June 2009.

[8] X. Pan, J. Xu, and X. Meng, "Protecting location privacy against location-dependent attacks in mobile services," *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, no. 8, pp. 1506–1519, 2012.

[9] B. Gedik and L. Liu, "Protecting location privacy with personalized k-anonymity: architecture and algorithms," *IEEE Transactions on Mobile Computing*, vol. 7, no. 1, pp. 1–18, 2008.

[10] L. Wang, D. Zhang, D. Yang, B. Y. Lim, X. Han, and X. Ma, "Sparse mobile crowdsensing with differential and distortion location privacy," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 2735–2749, 2020.

[11] C. Xu, L. Zhu, Y. Liu, J. Guan, and S. Yu, "Dp-ltod: Differential privacy latent trajectory community discovering services over location-based social networks," in *Proceedings of the IEEE Transactions on Services Computing*, p. 1, China, July 2018.

[12] J. Shao, R. Lu, and X. Lin, "Fine: A fine-grained privacy-preserving location-based service framework for mobile devices," in *Proceedings of the IEEE INFOCOM 2014 - IEEE Conference on Computer Communications*, pp. 244–252, Toronto, ON, USA, May 2014.

[13] H. Li, L. Sun, H. Zhu, X. Lu, and X. Cheng, "Achieving privacy preservation in wifi fingerprint-based localization," in *Proceedings of the IEEE INFOCOM 2014 - IEEE Conference on Computer Communications*, pp. 2337–2345, Toronto, ON, USA, May 2014.

[14] B. Li, R. Liang, D. Zhu, W. Chen, and Q. Lin, "Blockchain-based trust management model for location privacy preserving in vanet," in *Proceedings of the IEEE Transactions on Intelligent Transportation Systems*, pp. 1–11, China, November 2020.

[15] B. Luo, X. Li, J. Weng, J. Guo, and J. Ma, "Blockchain enabled trust-based location privacy protection scheme in vanet," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 2, pp. 2034–2048, 2020.

[16] J. Zhang, F. Yang, Z. Ma, Z. Wang, X. Liu, and J. Ma, "A decentralized location privacy-preserving spatial crowdsourcing for internet of vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, pp. 1–15, 2020.

[17] S. Zou, J. Xi, H. Wang, and G. Xu, "Crowdblps: a blockchain-based location-privacy-preserving mobile crowdsensing system," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 6, pp. 4206–4218, 2020.

[18] R. Agrawal, "Order preserving encryption for numeric data," in *Proceedings of the 2004 ACM SIGMOD International Conference on Management of Data (SIGMOD '04)*, Paris, France, June 2004.

[19] A. Boldyreva, N. Chenette, Y. Lee, and A. O'Neill, "Order-preserving symmetric encryption," in *Proceedings of the International Conference on Advances in Cryptology-Eurocrypt*, Zagreb, Croatia, May 2009.

[20] Y. Peng, H. Li, J. Cui, and J. Zhang, "hope: Improved order preserving encryption with the power to homomorphic operations of ciphertexts," *Science China*, vol. 6, p. 062101, 2017.

[21] L. Xiao and I. L. Yen, "Security analysis for order preserving encryption schemes," in *Proceedings of the 46th Annual Conference on Information Sciences & Systems*, Princeton, NJ, USA, September 2012.

[22] R. C. Merkle, "A certified digital signature," in *Proceedings of the Conference on the Theory and Application of Cryptology*, Houthalen, Belgium, April 1989.

[23] C. Liu, R. Ranjan, C. Yang et al., "MuR-DPA: top-down levelled multi-replica Merkle hash tree based secure public auditing for dynamic big data storage on cloud," *IEEE Transactions on Computers*, vol. 64, no. 9, pp. 2609–2622, 2015.

[24] P. Bai, W. Zhang, and L. I. Cong, *Using Rank-Based Merkle Hash Tree into Homomorphic Authentication during Cloud*, 2018.

[25] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in *Proceedings of the Advances in Cryptoloty-EUROCRYPT 2005*, Aarhus, Denmark, May 2005.

[26] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *Proceedings of the IEEE*

Symposium on Security & Privacy, Aarhus, Denmark, May 2007.

[27] K. Yang, X. Jia, and K. Ren, "Secure and verifiable policy update outsourcing for big data access control in the cloud," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 12, pp. 3461–3470, 2015.

[28] H. E. Gafif, N. Meddah, and A. Toumanari, *A Lightweight Ciphertext-Policy Attribute-Based Encryption for Fine-Grained Access Control*Springer, Berlin, Germany, 2018.

[29] A. Gorodetskiy, A. Serebryakov, A. Oracevic, R. Hussain, and S. M. Ahsan Kazmi, "Towards a secure and efficient location-based secret sharing protocol," in *Proceedings of the 2020 International Symposium on Networks, Computers and Communications (ISNCC)*, pp. 1–6, Quebec, Canada, October 2020.

[30] M. R. Nosouhi, S. Yu, M. Grobler, Q. Zhu, and Y. Xiang, "Blockchaincbased location proof generation and verification," in *Proceedings of the IEEE INFOCOM 2019 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pp. 1–6, Paris, France, May 2019.

[31] X. Yi, R. Paulet, E. Bertino, and V. Varadharajan, "Practical approximate k nearest neighbor queries with location and query privacy," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 6, pp. 1546–1559, 2016.

[32] R.-H. Hwang, Y.-L. Hsueh, J.-J. Wu, and F.-H. Huang, "SocialHide: a generic distributed framework for location privacy protection," *Journal of Network and Computer Applications*, vol. 76, pp. 87–100, 2016.

[33] X. Gong, X. Chen, K. Xing, D.-H. Shin, M. Zhang, and J. Zhang, "From social group utility maximization to personalized location privacy in mobile networks," *IEEE/ACM Transactions on Networking*, vol. 25, no. 3, pp. 1703–1716, 2017.

[34] Y. Qiu, Y. Liu, X. Li, and J. Chen, "A novel location privacy-preserving approach based on blockchain," *Sensors*, vol. 20, no. 12, 2020.

[35] Y. Zheng, X. Xie, and W. Y. Ma, "Geolife: A collaborative social networking service among user, location and trajectory," *Bulletin of the Technical Committee on Data Engineering*, vol. 33, no. 2, pp. 32–39, 2010.