Blind Filtering at Third Parties: An Efficient Privacy-Preserving Framework for Location-Based Services

Jing Chen, Kun He, Quan Yuan, Min Chen, *Senior Member, IEEE*, Ruiying Du, and Yang Xiang

Abstract—Location-based service (LBS) has gained increasing popularity recently, but protecting users' privacy in LBS remains challenging. Depending on whether a trusted third party (TTP) is used, existing solutions can be classified into: TTP-based and TTP-free. The former relies on a TTP for user privacy protection, which creates a single-point-failure and is thus impractical in reality. The latter does not require any TTP, but usually introduces redundant point-of-interest (POI) records in query result and thus incurs significant computation and communication costs on the user side, making them unsuitable for resource-constrained mobile devices. In this paper, we propose a novel framework to protect user privacy while ensuring efficiency. Our framework also uses redundant POI records to protect privacy against LBS provider but employs a semi-trusted third party, called proxy, to filter out redundant POI records. To protect privacy against proxy, we design a novel filtering protocol, Blind filter, to allow the proxy to filter out redundant encrypted POI records in a blind way. In comparison with existing solutions, our framework is not only resilient to dual identity attack, but also incurs lower communication and computation overhead. Comprehensive analysis and experiments show that our framework is secure and highly efficient in mobile environments.

Index Terms—Location-based service, location privacy, blind filter, dual identity attack

1 Introduction

With the explosive growth in location-aware mobile devices, Location-Based Service (LBS) [1] becomes increasing popular with a growing number of applications (e.g., Yelp and TripAdvisor). In a typical LBS application, an LBS Provider (LBSP) offers services to users upon receiving their location-based queries. For example, a user may query the restaurants within 2 miles of his current location, or an available parking lot next to a central business district. According to the report from Berg Insight,[1] the global LBS revenues are EUR 10.3 billion in 2014, and will reach EUR 34.8 billion in 2020.

While LBSs offer great convenience to daily life, they raise significant privacy concerns. Since a typical LBS query usually includes a user's identity, her/his location, and other information, disclosing such information to LBS providers facilitates user profiling [2], [3], [4], [5], [6], [7], [8]. For example, a malicious LBSP can infer personal habits and interests or track a target user from her/his location-based queries [9].

Depending on whether a Trusted Third Party (TTP) is employed, existing privacy-preserving mechanisms for LBSs can be classified into two categories [10], [11]: *TTP-based* and *TTP-free*. Most of existing k-anonymity-based schemes [12], [13] and their variants [14], [15], [16] belong to the TTP-based solutions. These solutions rely on a TTP server to construct an anonymous set based on users' original queries to ensure that the LBSP cannot distinguish target user from at least $k-1$ other users. The TTP server does not only know users' geographic positions, but also the query results from the LBSP. By compromising a TTP server, an adversary can access all the sensitive information of users. To avoid such single-point-failure caused by TTP, such as [17], [18], [19], [20], [21] have been proposed in the literature. These TTP-free approaches require either no third party server or only a *semi-trusted* one. However, most of these solutions require users to issue fake LBS queries or receive redundant LBS records, which incurs high communication and computation overhead on the user side, making them unsuitable for resource-constrained mobile devices.

To the best of our knowledge, FINE [22] is the most practical TTP-free solution for mobile devices in which users

1. http://www.berginsight.com/ShowReport.aspx?m_m=3&id=212

- *J. Chen is with the School of Cyber Science and Engineering, Wuhan University, Wuhan 430072, China, and the Science and Technology on Communication Security Laboratory, Chengdu 610041, China. E-mail: chenjing@whu.edu.cn.*
- *K. He is with the School of Cyber Science and Engineering, Wuhan University, Wuhan 430072, China. E-mail: milloglobe@gmail.com.*
- *Q. Yuan is with the Computer School, University of Texas-Permian Basin, Odessa, TX 79762. E-mail: dantes.yuan@gmail.com.*
- *M. Chen is with the School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan 430074, China, and the Wuhan National Laboratory for Optoelectronics, Wuhan 430074, China. E-mail: minchen@ieee.org.*
- *R. Du is with the Collaborative Innovation Center of Geospatial Technology, Wuhan 430072, China. E-mail: duraying@126.com.*
- *Y. Xiang is with the School of Information Technology, Deakin University, Australia, and the State Key Laboratory of Integrated Service Networks, Xidian University, Xi'an 710126, China. E-mail: yang.xiang@deakin.edu.au.*

Manuscript received 19 Aug. 2017; revised 26 Dec. 2017; accepted 20 Feb. 2018. Date of publication 1 Mar. 2018; date of current version 1 Oct. 2018. (Corresponding author: Kun He.)
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TMC.2018.2811481

TABLE 1
Comparison between ePriLBS and Other Schemes

Property		ePriLBS	[22][a]	[22][b]	[19][c]
Privacy-preserving against LBSP	Identity	√	√	√	√
	Position	√	√	√	√
	Query message	√	√	√	√
	Final result	√	√	√	√
Privacy-preserving against third party server	Identity	×	×	×	×
	Position	×	√	×	×
	Query message	√	√	×	×
	Final result	√	√	×	×
Communication efficiency		√	√	√	−
Efficient computation for user		√	√	√	−
Efficient computation for server		√	×	×	√

[a] *if no dual identity attack exists.*
[b] *if dual identity attack exists.*
[c] *items marked with "−" means they depend on the cache.*

will not send or receive any redundant data. Under FINE, the LBSP outsources its encrypted Point Of Interest (POI) dataset to a semi-trusted cloud server which takes over computation intensive tasks from the LBSP and users. Users retrieve the encrypted POI records that exactly satisfy their encrypted LBS queries from the cloud server, and then decrypt them using proper keys obtained from the LBSP. By decoupling the dataset and data access, location privacy is protected against both the LBSP and the cloud server.

We observe three limitations of FINE. First, storing an extra copy of POI dataset, even encrypted, on the cloud server introduce additional vulnerabilities. For example, the cloud server (i.e., the semi-trusted third party server) can launch *dual identity attack* [23], by registering as a regular user to and obtaining decryption keys from the LBSP to decrypt the encrypted POI records it receives. Then, the cloud server can infer the user's location and interest via the decrypted POI records that satisfy an LBS query; that is completely break the privacy guarantee of FINE. Second, the cloud server sustains high computation overhead. In particular, for each position in the LBS query range, it needs to examine every encrypted POI record in the dataset via expensive operations (i.e., exponentiation and pairing). Also, the cloud server must synchronize with the LBSP frequently to ensure data consistency. Moreover, since all POI records are encrypted, the cloud server cannot benefit from any query optimization technique [24]. Third, FINE only supports simple query involving a query position and a range, while many LBS applications require complex queries that involve keywords and other information, such as "restaurant" and "available parking lot". Therefore, designing a TTP-free, privacy-preserving LBS system suitable for resource-constrained mobile devices remains an open challenge.

In this paper, we propose ePriLBS, a novel efficient privacy-preserving location-based service framework. ePriLBS adopts a semi-trusted third party as in FINE, called proxy, to simultaneously protect users' privacy and ensure query efficiency. Instead of storing POI dataset at the proxy, we let the proxy construct an anonymous query with a region containing at least k users, and forward it to the LBSP. Note that the anonymous query is partial encrypted to prevent the proxy from learning users' interests. Once the proxy receives the (encrypted) query result from the LBSP, it filters out redundant POI records in a blind way. In particular, we design Blind filter, a novel filtering protocol based on homomorphic encryption [25], [26] and a lightweight randomization technique to prevent information leakage against both the LBSP and the proxy. Our main contributions are summarized as follows.

(1) To the best of our knowledge, ePriLBS is the first TTP-free solution for protecting user privacy in LBS that not only withstands dual identity attacks, but also improves efficiency for all parties involved. Table 1 compares our framework with the most related works in terms of desired properties discussed in Section 2.3.
(2) We formally define the privacy against semi-trusted third party server for the first time. Even though the privacy against LBSPs has been widely studied, there has been no formal definition for privacy against third party in LBS. Our definition allows formal proof of LBS system that employs semi-trusted third party.
(3) We prove the security of ePriLBS and thoroughly evaluate its performance via detailed experiments. The experimental results confirm that our framework is highly efficient and suitable for mobile environments.

The rest of this paper is organized as follows. In Section 2, we formalize our design fundamentals including the system model, threat model, and design goals. We present the preliminaries and the details of our framework in Sections 3 and 4, respectively. In what follows, we give the security analysis and performance evaluation in Sections 5 and 6, respectively. Section 7 reviews related works. Finally, we conclude the paper in Section 8.

2 Models and Design Goals

In this section, we introduce the system and threat models as well as our design goals.

2.1 System Model

Our system consists of three types of entities: an LBS provider, a set of proxies, and many users. To ease the presentation, our subsequent discussion focuses on a single proxy and one user as shown in Fig. 1.

The LBSP stores POI dataset and answers location-based queries from users. Each POI record can be represented as $((x, y), desc)$, where (x, y) is the POI's x and y coordinates, and $desc$ is related descriptive information, such as its category. The LBS query issued by the user is a triple $Q = (id, (x, y), (r, f))$, where id is the user's identity, (x, y) is the user's x and y coordinates, and (r, f) is the query message. The radius r defines a geographic range of the LBS query, and the predicate f specifies the additional properties that the returned POI records need to satisfy. For example, f can be "parking lot AND available". The query result R consists of the set of POI records that exactly satisfy Q. Proxies are typically deployed in existing network infrastructures, such as WiFi access points and cellular base stations. The proxy provides (free or paid) privacy-preserving services to users by processing and transmitting messages between the users in its region and the LBSP.

The high level interaction among the user, the proxy, and the LBSP is as follows. The user submits a partial encrypted query through the proxy, which in turn constructs an

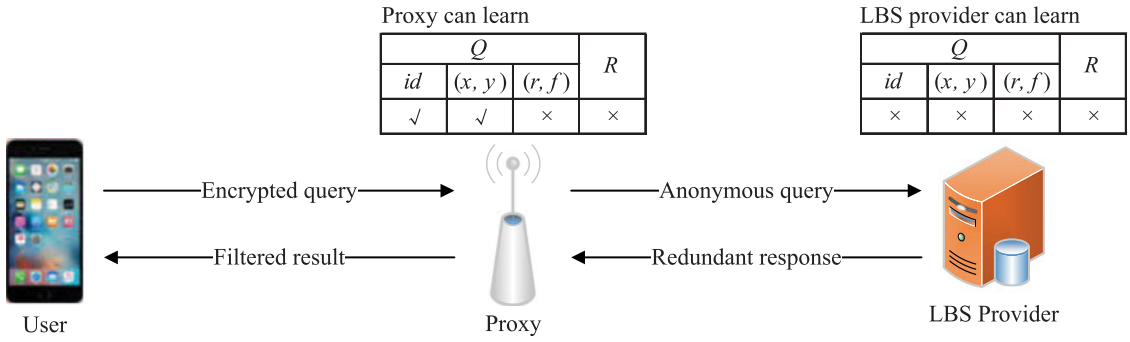


Fig. 1. The system model of our framework.

anonymous query from k user queries, and sends it to the LBSP. On receiving the anonymous query, the LBSP returns a response as the union of the results of all k user queries. The proxy then executes a blind filter protocol with the LBSP to filter out redundant POI records and returns accurate query result to each of the k users.

2.2 Threat Model

We assume that the communication channels between the LBSP and the proxy, and between the proxy and the user are secured by standard techniques, such as SSL/TLS and SSH, so that external adversaries cannot learn anything from the encrypted communications. In this paper, we focus on preventing information leakages at the semi-trusted LBSP and the proxy.

As in [27], [28], we assume that the LBSP is semi-trusted. Specifically, it is trusted to faithfully follow protocol execution but is interested in learning users' LBS queries and query results. As in [22], [29], we also assume that the LBSP cannot collude with the proxy. In addition, the LBSP can launch dual identity attack, in which the LBSP pretends to be a normal user and sends queries to the proxy to degrade anonymous query and extract users' queries. The dual identity attack is actually an active attack which looks contrary to the semi-trusted assumption. However, the adversary in the dual identity attack does not modify any part of the protocol in contrast with other types of active attackers, such as the man-in-the-middle attack. Note that, the dual identity attack is difficult to prevent or detect, but it is easy to implement in LBSs since the users can be anonymous.

A proxy is a semi-trusted party which faithfully follows protocol execution but may be interested in users' sensitive information [22], [23], [30], [31]. Since the proxy is normally deployed in existing infrastructures, such as WiFi access points and cellular base stations [19], [29], [32], it can always learn some information about users via physical channels. For instance, the proxy can identify users by their MAC addresses and estimate users' geographic positions by channel characteristics such as received signal strength. Therefore, we focus on preventing the proxy from learning the query messages (i.e., the radius and the predicate) and query results. Note that, although the POI dataset is public, the query results must be hidden from the proxy, otherwise the proxy can infer user's query messages from the query results. Moreover, the proxy may also launch dual identity attack, in which the proxy constructs the anonymous query based on its chosen position and query message to infer users' LBS queries.

2.3 Design Goals

We design ePriLBS with the following goals in mind.

- *Privacy-preserving:* The LBSP should not learn the LBS query or query result of individual user. Likewise, the proxy should not learn anything about the query message or query result of individual user.
- *Communication efficiency:* The protocol should be efficient in communication in the sense that the query result returned to the user should not contain any redundant POI records.
- *Computation efficiency:* The protocol should incur low computation overhead for all the parties involved.

3 Preliminaries

In this section, we briefly review some background of homomorphic encryption and k-anonymity.

3.1 Homomorphic Encryption

A homomorphic encryption scheme [33] $\mathcal{HE} = (\mathsf{HKeyGen}, \mathsf{HEnc}, \mathsf{HDec})$ allows specified computations on the ciphertexts without the need for decryption first. Specifically, for any public-private key pair (pk, sk) and any m_1, m_2 in the plaintext space, $\mathsf{HDec}_{sk}(\mathsf{HEnc}_{pk}(m_1) \otimes \mathsf{HEnc}_{pk}(m_2)) = m_1 \odot m_2$ holds, where $\otimes$ denotes the computation on the ciphertexts, and $\odot$ denotes the computation on the plaintexts.

In this paper, we use Paillier encryption scheme [34]. The details of this scheme are described as follows.

- $\mathsf{HKeyGen}$, the key generation algorithm, takes as input a security parameter, outputs a public key $pk_l = n$ and a private key $sk_l = s$, where $n := pq$ is the product of two prime p and q with equal length, and $s := (p-1)(q-1)$.
- HEnc, the encryption algorithm, takes as input a plaintext $m \in \mathbb{Z}_n$, outputs a ciphertext $c := (n+1)^m t^n \bmod n^2$, where $t \in \mathbb{Z}^*_{n^2}$ is a random integer.
- HDec, the decryption algorithm, takes as input a ciphertext c, outputs a plaintext $m := L(c^s \bmod n^2) \cdot s^{-1} \bmod n$, where $L(a) \stackrel{def}{=} (a-1)/\bmod n$.

The Paillier encryption scheme has two useful properties: $\mathsf{HDec}_{sk}(\mathsf{HEnc}_{pk}(m_1) \cdot \mathsf{HEnc}_{pk}(m_2)) = m_1 + m_2$ and $\mathsf{HDec}_{sk}(\mathsf{HEnc}_{pk}(m_1)^{m_2}) = m_1 \cdot m_2$.

3.2 k-Anonymity

ePriLBS relies on existing privacy-preserving technique to ensure user privacy against the LBSP. While ePriLBS can be built on top of many existing privacy-preserving techniques,

we take existing cloaking technique [19] as an example in this paper, which ensures every user is indistinguishable from the other $k-1$ users from the LBSP's perspective [35]. The value of k depends on desired privacy level, which is usually from 5 to 20 in the literature (e.g., [13], [19]).

Specifically, the cloaking algorithm takes as input k users' geographical positions $\{(x_1, y_1), (x_2, y_2), \ldots, (x_k, y_k)\}$, and outputs a rectangular area $rect$ which is a minimum boundary rectangle that contains these k locations. In case there are less than k user locations, dummy users can be inserted to construct the area. Note that, the straightforward use of the aforementioned cloaking algorithm may fail to achieve k-anonymity, since the LBSP can learn all users' locations if their locations are very close to each other. Another problem of the aforementioned cloaking algorithm is that it is vulnerable to background knowledge attacks in which the LBSP has some information about the locations and users' potential query messages. Fortunately, many cloaking algorithms can resist these kinds of attacks, such as [13] and [19]. The main idea of these clocking algorithms is to carefully choose some dummy locations and query messages, and mix the true locations and query messages with the dummy ones. In this paper, we employ this kind of cloaking algorithms with strong privacy guarantees.

4 ePriLBS Framework

In this section, we first give an overview of the ePriLBS framework. We then present a novel Blind Filter protocol and detail ePriLBS's design.

4.1 Overview

ePriLBS is designed to protect user privacy against both the LBSP and the proxy. Specifically, we use the traditional cloaking technique to protect user privacy against the LBSP, by which the proxy generates an anonymous query with a cloaking area that contains at least k users. To defend user privacy against the proxy, we encrypt the query message and the query result using a session key shared between the LBSP and the user.

Under the cloaking technique, the encrypted query result returned by the LBSP contains redundant POI records. To ensure efficiency on the user side, redundant POI records need be filtered at the proxy. Recall that each POI record consists of a geographic position (x_i, y_i) and associated description $desc_i$, and the query message consists of a radius r and a predicate f. The user's location is needed to generate cloaking area and usually in plaintext, while r and f are encrypted. The challenge in filtering redundant POI records is then how to allow the proxy to learn correct POI records without letting the proxy and the LBSP learn any information from this process. One may think that this challenge can be solved using homomorphic encryption. Unfortunately, it has been shown in [25] that directly applying homomorphic encryption would allow the LBSP to learn $d_i^2 - r^2$ from the ciphertext, where d_i is the distance between the POI record and the user's position, and r is the query radius. As a result, the LBSP can learn the POI records that satisfy $d_i = r$ and further compute the user's location via trilateration.

To tackle this challenge, we design a novel protocol called *Blind filter* that integrates a lightweight randomization technique with homomorphic encryption to allow the proxy to filter out redundant POI records without either the proxy or the LBSP violating user privacy. Table 2 summarizes the notations used in the ePriLBS framework.

TABLE 2
The Notations Used in the ePriLBS Framework

Notation	Description
$rect$	area that satisfies the k-anonymity requirement
n	product of two primes p and q
m	the bit length of the geographic coordinates where $m \ll \|n\|$
λ_a	security parameter used in the asymmetrical encryption scheme
(pk_a, sk_a)	LBSP's key pair of the asymmetrical encryption scheme
λ_h	security parameter used in the homomorphic encryption scheme
(pk_l, sk_l)	LBSP's key pair of the homomorphic encryption scheme
(pk_p, sk_p)	proxy's key pair of the homomorphic encryption scheme
κ, λ_κ	session key and its length
l, λ_l	random label and its length
c_κ, c_q	encrypted session key and query message
c, (c_0, c_1, c_2)	encrypted POI record and geographical position of the POI record
c_r, c_d	challenge sent from the proxy in the blind filter
c_p	response sent from the LBSP in the blind filter
δ	random nonzero integer between $-2^{(\|n\|-m-1)/2}+1$ and $2^{(\|n\|-m-1)/2}$
Δ	random integer between $-2^{m-1}+1$ and 2^{m-1}
δ'	random positive integer which is less than $2^{(\|n\|-m-1)/2}$

4.2 Blind Filter

As mentioned in Section 4.1, the core component in ePriLBS is the Blind filter protocol. In this section, we first give a formal definition of this protocol, and then propose a concrete construction.

The Blind filter protocol is executed between the LBSP and the proxy. Recall that a user's LBS query is $(id, (x, y), (r, f))$, and that the proxy and the LBSP learn $(id, (x, y))$ and (r, f), respectively. In our framework, the proxy constructs an area $rect$ that contains (x, y) via cloaking technique, and sends this area to the LBSP. Then, the LBSP can determine a subset $\mathcal{D}$ of its POI dataset, that every record in $\mathcal{D}$ is within an expanded area which is determined by $rect$ and r.

Definition 1 (Blind filter). *Let* $\mathsf{SEnc}(\cdot)$ *be a symmetric encryption algorithm. Blind filter is an interactive protocol between the LBSP and a proxy. The LBSP inputs a set of geographic positions* $\mathcal{D}$ *and a radius* r*, and obtains a set of encrypted positions* $\mathcal{C} = \{\mathsf{SEnc}((x_i, y_i)) \,|\, (x_i, y_i) \in \mathcal{D}\}$*. The proxy inputs a geographic position* (x, y)*, and obtains a set of encrypted positions* $\{\mathsf{SEnc}((x_i,\ y_i)) \in \mathcal{C} \,|\, \sqrt{(x-x_i)^2 + (y-y_i)^2} \le r\}$*. During the protocol, the LBSP could not be aware of which subset is selected by the proxy, and the proxy cannot learn* (x_i, y_i) *or* r*.*

Our realization of Blind filter is based on integrating homomorphic encryption and a lightweight randomization

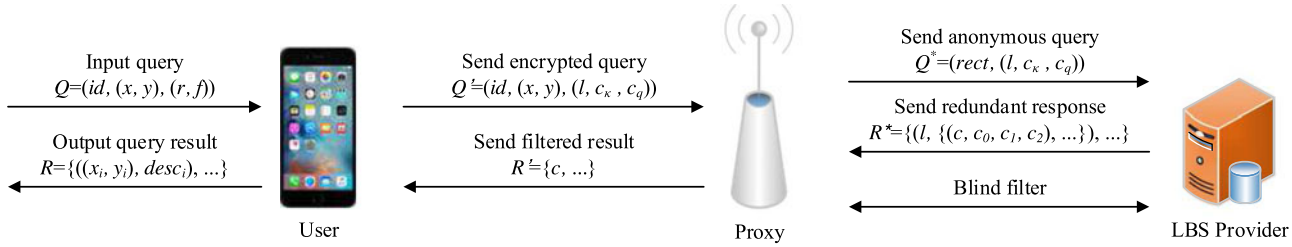


Fig. 2. The workflow of the ePriLBS framework.

approach specifically designed to address the limitation identified in [25]. In what follows, we first give an overview of our realization and then detail its construction.

We find that to determine whether a POI record is redundant while protecting user privacy against the LBSP and the proxy, the key challenge is to allow the proxy to learn whether d_i is smaller than r without the LBSP to learn this relationship, where $d_i = \sqrt{(x-x_i)^2+(y-y_i)^2}$. To tackle this challenge, we let the proxy to compute $d_i^2 - r^2$ in encrypted form. To prevent the LBSP from learning the relationship between d_i^2 and r^2, the proxy randomizes the encrypted $d_i^2 - r^2$ by a random affine transformation. To further prevent the proxy from learning the accurate value of $d_i^2 - r^2$, the LBSP randomizes the encrypted $d_i^2 - r^2$ by a random scale. Then, the proxy can only learn the relationship between d_i and r, while the LBSP learns nothing.

We now detail the Blind filter protocol, which consists of four stages: *setup*, *challenge*, *response*, and *output*. Let (pk_l, sk_l) be the public-private key pair of the LBSP, and (pk_p, sk_p) be the public-private key pair of the proxy.

Setup Stage. The LBSP processes as follows with a session key κ, the radius r, and each position (x_i, y_i) in $\mathcal{D}$.

- Encrypt the position (x_i, y_i) by running $\mathsf{SEnc}_\kappa((x_i, y_i))$ with the session key κ.
- Compute

$$\begin{aligned} c_0 &\leftarrow \mathsf{HEnc}_{pk_l}(x_i^2 + y_i^2 - r^2) \\ c_1 &\leftarrow \mathsf{HEnc}_{pk_l}(-2x_i) \\ c_2 &\leftarrow \mathsf{HEnc}_{pk_l}(-2y_i). \end{aligned} \tag{1}$$

- Send $C = (\mathsf{SEnc}((x_i, y_i)), c_0, c_1, c_2)$ to the proxy.

Challenge Stage. The proxy processes as follows with (x, y) and C.

- Choose a random nonzero integer $-2^{(|n|-m-1)/2}+1 \le \delta \le 2^{(|n|-m-1)/2}$ and a random integer $-2^{m-1}+1 \le \Delta \le 2^{m-1}$.
- Compute

$$\begin{aligned} c_r &\leftarrow \mathsf{HEnc}_{pk_l}(\Delta) \cdot \\ &\quad (\mathsf{HEnc}_{pk_l}(x^2+y^2) \cdot c_0 \cdot c_1^x \cdot c_2^y)^\delta \\ c_d &\leftarrow \mathsf{HEnc}_{pk_p}(-\Delta). \end{aligned} \tag{2}$$

- Send the challenge (c_r, c_d) to the LBSP.

Response Stage. The LBSP processes as follows.

- Choose a random positive integer $\delta' \le 2^{(|n|-m-1)/2}$.
- Compute

$$c_p \leftarrow (\mathsf{HEnc}_{pk_p}(\mathsf{HDec}_{sk_l}(c_r)) \cdot c_d)^{\delta'}. \tag{3}$$

- Send the response c_p to the proxy.

Output Stage. The proxy processes as follows.

- Accept $\mathsf{SEnc}((x_i, y_i))$, if $\mathsf{HDec}_{sk_p}(c_p) \cdot \delta \le 0$. Otherwise, reject it.

The value ranges of δ, Δ, and δ' ensure that the computations will not cause overflow. It is easy to prove the correctness of our Blind filter protocol. Let $d_i = \sqrt{(x-x_i)^2+(y-y_i)^2}$ be the distance between (x, y) and (x_i, y_i). We have

$$\begin{aligned} c_r &= (\mathsf{HEnc}_{pk_l}(x^2+y^2) \cdot \mathsf{HEnc}_{pk_l}(x_i^2+y_i^2-r^2) \\ &\quad \cdot \mathsf{HEnc}_{pk_l}(-2x_i)^x \cdot \mathsf{HEnc}_{pk_l}(-2y_i)^y)^\delta \cdot \mathsf{HEnc}_{pk_l}(\Delta) \\ &= \mathsf{HEnc}_{pk_l}(x^2+y^2+x_i^2+y_i^2-r^2-2xx_i-2yy_i)^\delta \\ &\quad \cdot \mathsf{HEnc}_{pk_l}(\Delta) \\ &= \mathsf{HEnc}_{pk_l}(\delta \cdot (d_i^2 - r^2) + \Delta) \end{aligned}$$

and

$$\begin{aligned} c_p &= (\mathsf{HEnc}_{pk_p}(\delta \cdot (d_i^2 - r^2) + \Delta) \cdot \mathsf{HEnc}_{pk_p}(-\Delta))^{\delta'} \\ &= \mathsf{HEnc}_{pk_p}(\delta\delta' \cdot (d_i^2 - r^2)). \end{aligned}$$

If $d_i \le r$, then $d_i^2 - r^2$ is not greater than 0. We thus have

$$\mathsf{HDec}_{sk_p}(c_p) \cdot \delta = \delta\delta' \cdot (d_i^2 - r^2) \cdot \delta = \delta^2\delta' \cdot (d_i^2 - r^2) \le 0,$$

which means that the proxy can determine whether a particular POI record is within a circle of radius r centered at (x, y).

4.3 ePriLBS Design

We now detail ePriLBS framework. Besides the tools described in Section 3, we also use a public-key encryption scheme such as RSA, and a symmetric encryption scheme such as AES in our framework. The key generation algorithm, the encryption algorithm, and the decryption algorithm of the public-key encryption scheme are denoted by $\mathsf{PKeyGen}$, PEnc, and PDec, respectively. The encryption algorithm and the decryption algorithm of the symmetric encryption scheme are denoted by SEnc and SDec, respectively. We also extend the Blind filter protocol in Section 4.2 to POI record which contains a point and an additional description as shown in Section 2.1. The ePriLBS framework consists of six phases: System Initialization, Query Generation, Query Process, Data Retrieval, Response Filtering, and Result Recovery. The workflow of our framework is shown in Fig. 2.

4.3.1 System Initialization

In this phase, the LBSP generates its public-private key pairs via the *LBSP initialization stage*. When a proxy intends to register in the system, it runs the *proxy initialization stage*.

TABLE 3
The Query Information Table

User ID	Position	Label
id_1	(x_1, y_1)	l_1
id_2	(x_2, y_2)	l_2
...	...	...
id_k	(x_k, y_k)	l_k

LBSP Initialization Stage. According to the security parameters λ_a and λ_h, the LBSP executes following operations.

- Generate a key pair $(pk_a, sk_a) \leftarrow \mathsf{PKeyGen}(1^{\lambda_a})$ of the public-key encryption scheme.
- Generate a key pair $(pk_l, sk_l) \leftarrow \mathsf{HKeyGen}(1^{\lambda_h})$ of the homomorphic encryption scheme.
- Publish the public keys pk_a and ph_l, and keep sk_a and sk_l secret.

When a user or a proxy joins the system, he obtains the public keys of the LBSP.

Proxy Initialization Stage. According to the security parameter λ_h, the proxy executes following instructions.

- Generate a key pair $(pk_p, sk_p) \leftarrow \mathsf{HKeyGen}(1^{\lambda_h})$ of the homomorphic encryption scheme.
- Publish the public key pk_p, and keep the private key sk_p secret.

Note that users do not generate any key in this phase. In addition, if some proxies need be revoked, traditional approaches such as Certificate Revocation List can be used.

4.3.2 Query Generation

In this phase, a user generates an encrypted location-based query Q'. Suppose that the original LBS query is $Q = (id, (x, y), (r, f))$. The user executes the following operations.

- Generate a session key $\kappa \leftarrow \{0,1\}^{\lambda_\kappa}$, where λ_κ is the bit length of the key.
- Choose a random label $l \leftarrow \{0,1\}^{\lambda_l}$, where λ_l is the bit length of the label.
- Encrypt the session key $c_\kappa \leftarrow \mathsf{PEnc}_{pk_a}(\kappa)$ under the public key pk_a of the LBSP.
- Encrypt the query message $c_q \leftarrow \mathsf{SEnc}_\kappa((r, f))$ under the session key.
- Send the encrypted query $Q' = (id, (x, y), (l, c_\kappa, c_q))$ to the proxy.

4.3.3 Query Process

In this phase, the proxy constructs an anonymous query Q^* from k encrypted queries $Q'_1, \ldots, Q'_k$, where $Q'_j = (id_j, (x_j, y_j), (l_j, c_{\kappa_j}, c_{q_j}))$ for all $j \in [1, k]$. A proxy runs the following operations with a set of encrypted queries.

- Initialize an empty query information table as shown in Table 3.
- Construct a k-anonymity rectangle area $rect$ that contains k users.
- Insert query information of each user in the k-anonymity area into the query information table.
- Send anonymous query $Q^* = (rect, \{(l_j, c_{\kappa_j}, c_{q_j})\}_{j=1}^{k})$ to the LBSP.

In some situations, users need rapid response and there are not enough users for constructing k-anonymity area. A proxy can then generate fake queries to achieve k-anonymity [19].

4.3.4 Data Retrieval

In this phase, the LBSP searches appropriate POI records and encrypts them under the session key. Then, it sends all the encrypted POI records to the proxy. The LBSP processes each $(l_j, c_{\kappa_j}, c_{q_j})$ in Q^* as follows.

- Decrypt the session key $\kappa \leftarrow \mathsf{PDec}_{sk_a}(c_{\kappa_j})$ using its private key sk_a.
- Decrypt the query message $(r, f) \leftarrow \mathsf{SDec}_\kappa(c_{q_j})$ using the session key κ.
- Search all appropriate POI records for $(rect, (r, f))$ in the POI dataset.
- For each satisfied POI record $((x_i, y_i), desc_i)$, compute $c \leftarrow \mathsf{SEnc}_\kappa(((x_i, y_i), desc_i))$, and calculate corresponding (c_0, c_1, c_2) as in Equation (1).

After processing every $(l_j, c_{\kappa_j}, c_{q_j})$ in the anonymous query Q^*, the LBSP sends the response $R^* = \{(l_j, \{(c, c_0, c_1, c_2)_\zeta\}_{\zeta=1}^{\nu})\}_{j=1}^{k}$ to the proxy, where ν is the number of POI records that satisfies $(rect, (r, f))$.

Note that computing c_0, c_1, and c_2 can be accelerated by pre-computing $\mathsf{HEnc}_{pk_l}(x_i^2 + y_i^2)$, $\mathsf{HEnc}_{pk_l}(-2x_i)$, and $\mathsf{HEnc}_{pk_l}(-2y_i)$. Then, the LBSP does not need to compute c_1 and c_2 during the data retrieval phase. For computing c_0, the LBSP only needs to compute $\mathsf{HEnc}_{pk_l}(-r^2)$ once. Then, c_0 can be obtained by $\mathsf{HEnc}_{pk_l}(x_i^2 + y_i^2)$ and $\mathsf{HEnc}_{pk_l}(-r^2)$ via lightweight homomorphic operations. Thus, for each $(l_j, c_{\kappa_j}, c_{q_j})$ in the anonymous query, the LBSP only needs to run the time-consuming operation (i.e., the encryption algorithm) once no matter how many POI records satisfy it.

4.3.5 Response Filtering

In this phase, the proxy generates filtered result R' using the Blind filter protocol. Then, the proxy sends the filtered result to corresponding user. The proxy who fetches $(l_j, \{(c, c_0, c_1, c_2)_\zeta\}_{\zeta=1}^{\nu})$ in the set R^* does the following.

- Search the entry $(id, (x, y), l)$ in the query information table with the label matching the label in $(l_j, \{(c, c_0, c_1, c_2)_\zeta\}_{\zeta=1}^{\nu})$.
- For each (c, c_0, c_1, c_2), choose two random integers δ and Δ and compute c_r and c_d as in Equation (2).
- Send the challenge $\{(c_r, c_d)_\zeta\}_{\zeta=1}^{\nu}$ to the LBSP.

On receiving the challenge, the LBSP does the following.

- For each (c_r, c_d), choose a random positive integer δ', and computes c_p as in Equation (3).
- Send the response $\{(c_p)_\zeta\}_{\zeta=1}^{\nu}$ to the proxy.

After receiving the response $\{(c_p)_\zeta\}_{\zeta=1}^{\nu}$, the proxy does the following.

- Initialize an empty filtered result R'.
- For each c_p and corresponding c, δ, insert c into R' if $\mathsf{HDec}_{sk_p}(c_p) \cdot \delta \leq 0$.
- Send R' to the user whose identity is id, and remove corresponding row in the query information table.

Note that computation can be accelerated as shown in the data retrieval phase.

4.3.6 Result Recovery

In this phase, the user recovers the query result R from the filtered result R'. When the user receives R' from the proxy, he decrypts R' with the session key κ as $R \leftarrow \mathsf{SDec}_{\kappa}(R')$.

5 Analysis

In this section, we first analyze the correctness of proposed ePriLBS framework, and then examine the user privacy against the LBSP and the proxy, respectively. We also give a formal definition of privacy against the semi-trusted proxy.

5.1 Correctness

For each query message (r, f), the LBSP fetches all appropriate records based on the rectangle $rect$. Then, the query result to $((x, y), (r, f))$ is covered by the response R^*. As shown in Section 4.2, the Blind filter protocol outputs all encrypted records that satisfy (x, y) and r, and only outputs the satisfied records. Thus, the proxy can output the result that exactly satisfies the original query, which means ePriLBS framework achieves accurate query result.

5.2 Privacy against the LBSP

We adopt the k-anonymity definition for privacy against the semi-trusted LBSP [19]. To measure the privacy offered by k-anonymity, entropy-based metric is defined as follows. The entropy H of identifying the location of target user out of the anonymous query is defined as $H = -\sum_{i=1}^{k} p_i \log(p_i)$, where p_i denotes the probability that the ith query message belongs to the target user. When all p_i has probability $1/k$, the entropy achieves maximum.

Since the anonymous query is constructed via cloaking technique, such as [19], the ePriLBS framework provides entropy H as the underlying cloaking technique does, which means that the LBSP cannot identify the target user from the anonymous query. The only difference between our framework and other k-anonymity-based schemes is that the LBSP runs Blind filter protocol in the ePriLBS framework. More precisely, in ePriLBS, the LBSP knows all POI records that satisfy the query $(rect, (r, f))$, and receives c_r and c_d in the response filtering phase. Recall that $c_r = \mathsf{HEnc}_{pk_l}(\delta \cdot (d_i^2 - r^2) + \Delta)$ is a ciphertext under the public key of the LBSP, and c_d is a ciphertext under the public key of the proxy. Since the homomorphic encryption scheme is secure, the LBSP cannot learn anything from c_d. That means, the LBSP can only obtain $\delta \cdot (d_i^2 - r^2) + \Delta$ via the response filtering phase by decrypting c_r. However, δ and Δ are two random integers, which obfuscate the relationship between d_i and r. Specifically, the LBSP cannot distinguish the following three cases: 1) $d < r$, 2) $d = r$, and 3) $d > r$. Thus, the LBSP cannot learn anything from the response filtering phase, which means our framework can protect both original queries and query results against the LBSP, as the underlying cloaking technique does [19].

We then consider the case that the LBSP runs the dual identity attack. As we discussed, since the LBSP cannot learn any information from the response filtering phase, our framework provides the same privacy protection against the LBSP as the underlying cloaking technique does. Thus, if the cloaking technique employed in our framework can resist the dual identity attack (e.g., [13] can resist this attack), the ePriLBS framework is secure against such attack.

Note that, the cloaking operation executed by the proxy may fail if all k users are at similar location. In this case, we can employ other techniques to process users' positions, such as location-label based approaches [36].

5.3 Privacy against the Proxy

To the best of our knowledge, there has been no formal definition of the privacy against the semi-trusted proxy in LBS. We define the privacy against a semi-trusted proxy by the following game, where the proxy acts as an adversary.

(1) The proxy chooses an identity id and a geographic position (x, y) and sends them to the user.
(2) The proxy does the following for a polynomial number of times.
  (a) The proxy chooses a query message (r, f) and sends it to the user, which in turn generates the encrypted query from $(id, (x, y), (r, f))$.
  (b) On receiving the encrypted query, the proxy generates an anonymous query and sends it to the LBSP.
  (c) The proxy executes the Blind filter protocol with the LBSP and obtains filtered result.
(3) The proxy generates two distinct queries $Q_0 = (id, (x, y), (r_0, f_0))$ and $Q_1 = (id, (x, y), (r_1, f_1))$, and sends them to the user. We require that the redundant response and filtered result of Q_0 and Q_1 contain the same number of POI records, otherwise the proxy can distinguish Q_0 and Q_1 via the number of POI records. We also assume that the length of each encrypted POI record is identical for the same reason, which can be achieved by padding all POI records to the same length.
(4) The user chooses a random bit $b \in \{0, 1\}$, and sends encrypted query Q' to the proxy, that Q' is generated from Q_b. The proxy generates an anonymous query for Q', sends the anonymous query to the LBSP, and obtains filtered result from the redundant response.
(5) The proxy outputs a bit b'. The game returns 1 if $b' = b$, and 0 otherwise.

Definition 2 (Privacy against the proxy). *A scheme is secure against the semi-trusted proxy, if for any Probabilistic Polynomial Time (PPT) proxy, the probability that the game outputs 1 is negligible greater than 1/2.*

Since the proxy can always randomly guess, we define the system is secure against semi-trusted proxies if the game cannot outputs 1 with probability non-negligibly greater than 1/2. Definition 2 covers various existing attacks, including the dual identity attack in Section 2.2. To see that, the proxy can generate $k - 1$ encrypted queries when it constructs an anonymous query.

Theorem 1. *The ePriLBS framework is secure against the semi-trusted proxies, if the symmetric encryption scheme, the asymmetric encryption scheme, and the homomorphic encryption scheme are secure.*

TABLE 4
The Notations Used in the Performance Evaluation

Notation	Description
N_D	the number of POI records in the database
N_R	the number of POI records satisfying the original query
N_Q	the number of POI records satisfying the anonymous query
N_P	the number of points in the query range
N_S	the number of attributes in the system
T_s	the time of symmetric encryption and decryption
T_l	the time of location-based search
T_m	the time of multiplication
T_e	the time of exponentiation
T_p	the time of pairing
$\|E\|$	the size of encrypted POI record
$\|Z\|$	the element size in $\mathbb{Z}$
$\|G\|$	the element size in $\mathbb{G}$

Proof. Recalling the game in Definition 2, the proxy can obtain $(l_j, c_{\kappa_j}, c_{q_j})$, $(l_j, \{(c, c_0, c_1, c_2)_\zeta\}_{\zeta=1}^{\nu})$, and $\{(c_p)_\zeta\}_{\zeta=1}^{\nu}$ when a user submits Q' to the proxy. First, l_j is a random label which is independent with the query message msg_b and corresponding query result, therefore, the proxy cannot obtain any information from l_j. Second, c_{q_j} and c are ciphertexts under a random session key. Since the symmetric encryption is secure, the proxy cannot learn anything about msg_b and corresponding query result, even when it knows some relationships between other msg and corresponding query results via Step 2 in the game. Third, c_{κ_j} and (c_0, c_1, c_2) are ciphertexts under the public keys of the LBSP. Again, the proxy cannot infer any information, since the asymmetric encryption scheme and the homomorphic encryption scheme are secure. Finally, the proxy can decrypt c_p and obtain $\delta' \cdot (d_i^2 - r^2)$, where δ' is a random number chosen by the LBSP. However, the proxy cannot solve r or (x_i, y_i) from these numbers. Thus, the construction is secure against the semi-trusted proxy.

More formally, we can construct a series of games by replacing c_{q_j}, c, c_{κ_j}, and (c_0, c_1, c_2) with random values step by step. Then, the probability that the proxy can distinguish these games is negligible. In the last game, since c_p is computed by random (c_0, c_1, c_2), the probability that the proxy wins is exactly $1/2$. Thus, the probability that the proxy breaks Definition 2 is negligible greater than $1/2$. □

6 Performance Evaluation

Due to both ePriLBS and FINE are TTP-free schemes in which the computation and communication on the user side are efficient (other TTP-free schemes do not achieve such efficiency on user side), we compare our framework with the FINE framework [22] in terms of theoretical comparison and experimental performance, respectively, in this section.

TABLE 6
The Communication Comparison with FINE

	TYPE I	TYPE II
ePriLBS	$N_R \cdot \|E\|$	$N_Q \cdot (\|E\| + 6\|Z\|)$
FINE [22]	$N_R \cdot (\|E\| + 3\|G\|)$	0

6.1 Theoretical Comparison

In this section, we give theoretical comparison between ePriLBS and FINE. The notations used in comparisons are shown in Table 4.

Table 5 compares the computation cost. We omit some constant cost in both frameworks for simplicity. The computation cost on the user side in both frameworks only depends on the number of POI records N_R satisfying the original query. That means both frameworks achieve computation efficiency on the user side. However, our framework is more efficient, since for each encrypted POI record in the filtered result, the user in FINE has to compute the decryption key before decrypting it.

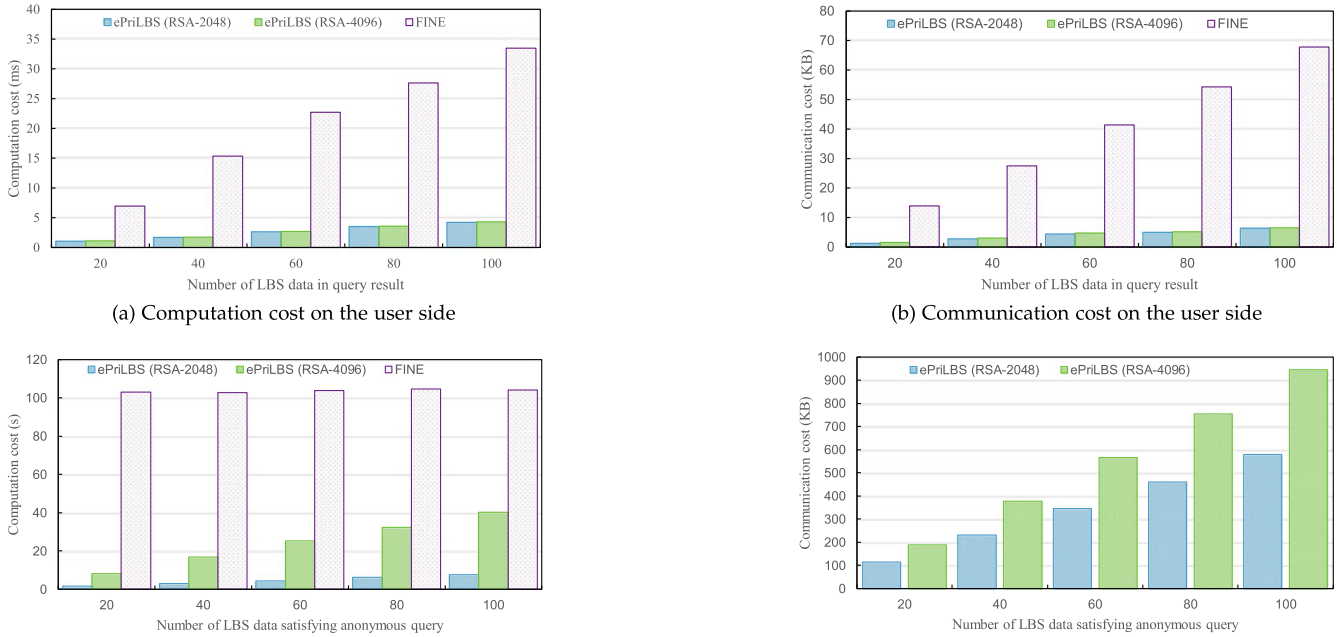
The computation cost on the server (the third party server and the LBSP) side is more complex, but they can be divided into two stages: search and process. The computation cost in search stage (i.e., search suitable POI records) depends on the number of POI records N_D in the database. In FINE, for each geographic point in the query range, the third party has to test every encrypted POI record in the database, and the computation cost for testing grows linearly with the number of attributes N_S. Thus, the computation complexity in search stage is $O(N_D \cdot N_P \cdot N_S)$, and the operations (i.e., pairing) in search stage is time-consuming. In contrast, the computation complexity of our framework in search stage is at most $O(N_D)$, and there is no time-consuming operation in this stage. In process stage (i.e., process suitable POI records), the computation cost (pairing, exponentiation, and multiplication) grows linearly with N_R in FINE, while the computation cost (exponentiation and multiplication) grows linearly with the number of POI records N_Q satisfying the anonymous query in our framework. However, since $N_R \cdot T_p \approx N_Q \cdot T_e$ in practice, our framework is as efficient as FINE in process stage.

Table 6 compares the communication cost. TYPE I means the communication cost between the user and the third party server, and TYPE II presents the communication cost between the third party server and the LBSP. Again, we omit some constant cost in both frameworks for simplicity. The communication cost on the users side (i.e., TYPE I) in both frameworks only depends on N_R. That means both frameworks achieve communication efficiency. However, the user in FINE has to receive a decryption key for each POI record in the query result. Therefore, the communication cost in FINE is larger.

Due to Blind filter, the communication cost on the server side (i.e., TYPE II) in our framework is larger than FINE.

TABLE 5
The Computation Comparison with FINE

	User	Third party server	LBSP
ePriLBS	$N_R \cdot T_s$	$10N_Q \cdot T_m + 10N_Q \cdot T_e$	$N_Q \cdot T_s + 8N_Q \cdot T_m + 10N_Q \cdot T_e + N_D \cdot T_l$
FINE [22]	$N_R \cdot T_s + N_R \cdot T_m + N_R \cdot T_e$	$(2N_DN_PN_S + 2N_RN_S + 2N_DN_P) \cdot T_m + (4N_DN_P + 3N_R) \cdot T_e + (4N_DN_PN_S + 4N_RN_S) \cdot T_p$	0



(a) Computation cost on the user side

(b) Communication cost on the user side

(c) Computation cost on the server side

(d) Communication cost on the server side

Fig. 3. Experimental results of ePriLBS and FINE in terms of computation and communication costs.

However, this extra communication cost is acceptable for the servers. Also it is worthwhile, since increasing the communication cost on the server side can significantly reduce the computation and communication costs on the user side.

6.2 Experimental Performance

The ePriLBS framework is implemented by OpenSSL 1.0.1 on a computer with Intel 3.2 GHz CPU. We also implement FINE [22] with OpenSSL 1.0.1 and PBC library 0.5.14 (with Type A pairings). The symmetric encryption and public-key encryption are AES-256-CBC and RSA-2048/RSA-4096, respectively. The homomorphic encryption is also based on RSA-2048/RSA-4096. We choose a POI dataset which contains 486822 POI records for following experiments, and the radius of query is 1 km.

Fig. 3a presents the computation cost on the user side. In both ePriLBS and FINE, the computation cost for generating the encrypted query is constant, and the computation cost for obtaining the query result grows linear with the number of POI records in the query result. In ePriLBS, users only need to perform the time-consuming algorithm once, i.e., in the query generation phase. Since users have to carry out exponentiations before decrypting encrypted POI records for every item of the query result in FINE, ePriLBS is more efficient than FINE.

Fig. 3b shows the communication cost on the user side. In both ePriLBS and FINE, the sizes of data sent by users are constant, and the sizes of data that users receive grow linearly with the number of POI records in query result. However, users only receive encrypted POI records in ePriLBS, while three extra elements of group should be received for each item of the query result in FINE. Thus, the communication cost in ePriLBS is more efficient in practice.

We accelerate computations on the server side in both ePriLBS and FINE by pre-computation. Note that additional storage cost for acceleration grows linearly with N_D and $N_D \cdot N_U$ in ePriLBS and FINE, respectively, where N_D is the number of POI records in the dataset, and N_U is the number of users in the system. However, the computation time for searching appropriate POI records is large in FINE which is almost 5 ms for each test, while searching the entire database which contains 486822 POI records spends less than 1 second in ePriLBS. Thus, we reduce the size of LBS database to 1000 for comparison, and decrease the number of points in the query range in FINE to 100 (which should be $1000 \cdot 1000$ since the unit is 1 meter). Fig. 3c shows the computation cost on the server side. ePriLBS is more practical than FINE due to the difference of computation cost in searching. Although RSA-2048 is secure enough in at present, the efficiency of using RSA-4096 in ePriLBS is still significantly better than FINE.

Finally, Fig. 3d shows the communication cost on the server side (i.e., between the LBSP and a proxy). There is no communication cost on the server side in FINE. The communication cost in ePriLBS is determined by the homomorphic encryption algorithm used in Blind filter. However, since the LBSP and proxies usually communicate via high-speed channels, the communication cost (less than 1 MB when using RSA-4096 based homomorphic encryption algorithm) in ePriLBS is acceptable in practice.

7 Related Work

A number of privacy-preserving techniques have been proposed to protect users' privacy in LBS [10], [11]. Based on whether a Trusted Third Party is employed, current solutions can be divided into two main categories: *TTP-based* and *TTP-free*. In this paper, we focus on the techniques that protect users' location privacy via position manipulation; therefore, pseudonym-based solutions [37], [38], [39] are beyond the scope of this paper.

TTP-Based Solutions. The most common TTP-based solutions are built on the cloaking technique which was introduced into LBS by Gruteser and Grunwald [12]. As an

entity between the LBSP and users, a TTP server receives a user's LBS queries which includes her/his sensitive information (e.g., user's identity and geographic location), and then blurs them by constructing new LBS queries with the quires of other $k-1$ real or dummy users. The new disguised queries are then sent to the LBSP to request services. Meanwhile, the TTP server needs to maintain all the original LBS queries, in order to resolve the correct results when it obtains any response from the LBSP. By employing the k-anonymity technique, it can safeguard location privacy against malicious LBSPs, and minimize the computation and communication cost on the user side. To enhance the indistinguishability of the cloaking technique, it also combines with other techniques, such as l-diversity [14], game theory [15], and so on [13], [35], [40]. However, since TTP servers know too much sensitive information of users, they may become the security bottleneck of the LBS applications.

TTP-Free Solutions. To avoid the leakage risk caused by TTP servers, researchers proposed other techniques to reduce TTP's necessity in LBSs, such as dummy location [19], [41], obfuscation [17], [42], [43], [44], [45]. Solutions in [1], [18] introduced a cryptographic technique, Private Information Retrieval (RIP), to achieve private retrieval on public database into LBSs. Geometric-based technique [46], [47] and differential privacy technique [28], [48], [49] also did not rely on TTP servers to protect location privacy. Unfortunately, all these solutions cause significant computation cost on the user side either when constructing LBS queries or when processing redundant POI records, which is unaffordable to the mobile devices. To design a privacy-preserving solution in mobile environments, Shao et al. [22] presented framework FINE, which relies on a semi-trusted third party that acts like a virtual provider. Data transferred between the semi-trusted third party and users are encrypted under the public key of the LBSP, to prevent leakage on the third party. Users can acquire decryption key from the LBSP for decrypting the encrypted POI records. However, FINE is vulnerable to dual identity attack on the semi-trusted third party. Also, it brings unnecessary cost on the third party, and only supports simple queries, instead of complex ones, which are used in most current applications. Zhu et al. [50] proposed a similar solution, called EPQ, in which all LBS data is outsourced to a semi-trusted cloud server. Since the security of EPQ depends on a secret key that is only known by the LBSP and registered users, EPQ is also vulnerable to dual identity attack on the semi-trusted third party.

8 Conclusion

In this paper, we have proposed an efficient privacy-preserving framework for location-based services, named ePriLBS, which adopts a semi-trusted third party, called proxy. By designing and exploiting Blind filter, a novel filtering protocol, ePriLBS preserves users' privacy against both the LBSP and the proxy, while the computation and communication cost on the user side is kept efficient. Specifically, our solution not only enhances system security by resisting dual identity attack, but also improves efficiency in terms of the computation and communication cost on all parties in LBSs. Comprehensive analysis and experiments show that the ePriLBS framework is suitable in mobile environments.

Acknowledgments

This research was supported in part by the Key Laboratory of Aerospace Information Security and the Trusted Computing, Ministry of Education; by the National Natural Science Foundation of China under grants 61572380, 61772383, 61702379, 61772405, and 61572220; and by the Major State Basic Research Development Program of China under grant 2014CB340600.

References

[1] R. Paulet, M. Kaosar, X. Yi, and E. Bertino, "Privacy-preserving and content-protecting location based queries," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 5, pp. 1200–1210, May 2014.

[2] X. Chen, J. Pang, and R. Xue, "Constructing and comparing user mobility profiles for location-based services," in *Proc. 28th Annu. ACM Symp. Appl. Comput.*, 2013, pp. 261–266.

[3] J. Zhang, C. Chen, Y. Xiang, W. Zhou, and Y. Xiang, "Internet traffic classification by aggregating correlated naive bayes predictions," *IEEE Trans. Inf. Forensics Secur.*, vol. 8, no. 1, pp. 5–15, Jan. 2013.

[4] J. Zhang, Y. Xiang, Y. Wang, W. Zhou, Y. Xiang, and Y. Guan, "Network traffic classification using correlation information," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 1, pp. 104–117, Jan. 2013.

[5] Z. Wang, D. Zhang, X. Zhou, D. Yang, Z. Yu, and Z. Yu, "Discovering and profiling overlapping communities in location-based social networks," *IEEE Trans. Syst. Man Cybern.: Syst.*, vol. 44, no. 4, pp. 499–509, Apr. 2014.

[6] A. Gervais, H. Ritzdorf, M. Lucic, V. Lenders, and S. Capkun, "Quantifying location privacy leakage from transaction prices," in *Proc. Eur. Symp. Res. Comput. Secur.*, 2016, pp. 382–405.

[7] S. Zhao, X. Luo, B. Bai, X. Ma, W. Zou, X. Qiu, and M. H. Au, "I Know Where You All Are! exploiting mobile social apps for large-scale location privacy probing," in *Proc. Australasian Conf. Inf. Secur. Privacy*, 2016, pp. 3–19.

[8] W. Luo, Y. Lu, D. Zhao, and H. Jiang, "On location and trace privacy of the moving object using the negative survey," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 1, no. 2, pp. 125–134, Apr. 2017.

[9] A. Beresford and F. Stajano, "Location privacy in pervasive computing," *IEEE Pervasive Comput.*, vol. 2, no. 1, pp. 46–55, Jan.–Mar. 2003.

[10] J. Krumm, "A survey of computational location privacy," *Pers. Ubiquitous Comput.*, vol. 13, no. 6, pp. 391–399, 2009.

[11] M. Wernke, P. Skvortsov, F. Dürr, and K. Rothermel, "A classification of location privacy attacks and approaches," *Pers. Ubiquitous Comput.*, vol. 18, no. 1, pp. 163–175, 2014.

[12] M. Gruteser and D. Grunwald, "Anonymous usage of location-based services through spatial and temporal cloaking," in *Proc. 1st Int. Conf. Mobile Syst., Appl. Serv.*, 2003, pp. 31–42.

[13] B. Niu, Q. Li, X. Zhu, G. Cao, and H. Li, "Achieving k-anonymity in privacy-aware location-based services," in *Proc. INFOCOM*, 2014, pp. 754–762.

[14] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam, "L-diversity: Privacy beyond k-anonymity," in *Proc. 22nd Int. Conf. Data Eng.*, 2006, pp. 24–24.

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

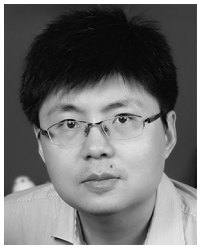
[16] T. Dargahi, M. Ambrosin, M. Conti, and N. Asokan, "ABAKA: A novel attribute-based k-anonymous collaborative solution for LBSs," *Comput. Commun.*, vol. 85, no. Supplement C, pp. 1–13, 2016.

[17] M. Duckham and L. Kulik, "A formal model of obfuscation and negotiation for location privacy," in *Proc. 3rd Int. Conf. Pervasive Comput.*, 2005, pp. 152–170.

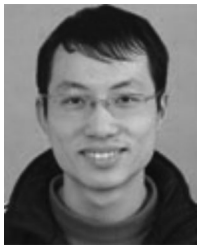
[18] G. Ghinita, P. Kalnis, A. Khoshgozaran, C. Shahabi, and K.-L. Tan, "Private queries in location based services: Anonymizers are not necessary," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 2008, pp. 121–132.

[19] B. Niu, Q. Li, X. Zhu, G. Cao, and H. Li, "Enhancing privacy through caching in location-based services," in *Proc. IEEE Conf. Comput. Commun.*, 2015, pp. 1017–1025.

[20] W. Ni, M. Gu, and X. Chen, "Location privacy-preserving k nearest neighbor query under users preference," *Knowl.-Based Syst.*, vol. 103, pp. 19–27, 2016.
[21] K. Olejnik, I. Dacosta, J. S. Machado, K. Huguenin, M. E. Khan, and J. P. Hubaux, "SmarPer: Context-aware and automatic runtime-permissions for mobile devices," in *Proc. IEEE Symp. Secur. Privacy*, 2017, pp. 1058–1076.
[22] J. Shao, R. Lu, and X. Lin, "FINE: A fine-grained privacy-preserving location-based service framework for mobile devices," in *Proc. INFOCOM*, 2014, pp. 244–252.
[23] Q. Wang, S. Hu, K. Ren, M. He, M. Du, and Z. Wang, "CloudBI: Practical privacy-preserving outsourcing of biometric identification in the cloud," in *Proc. Eur. Symp. Res. Comput. Secur.*, 2015, pp. 186–205.
[24] A. Aji, F. Wang, H. Vo, R. Lee, Q. Liu, X. Zhang, and J. Saltz, "Hadoop GIS: A high performance spatial data warehousing system over MapReduce," *Proc. VLDB Endowment*, vol. 6, no. 11, pp. 1009–1020, 2013.
[25] X.-Y. Li and T. Jung, "Search me if you can: Privacy-preserving location query service," in *Proc. INFOCOM*, 2013, pp. 2760–2768.
[26] S. Hu, Q. Wang, J. Wang, Z. Qin, and K. Ren, "Securing SIFT: Privacy-preserving outsourcing computation of feature extractions over encrypted image data," *IEEE Trans. Image Process.*, vol. 25, no. 7, pp. 3411–3425, Jul. 2016.
[27] X. Gong, X. Chen, K. Xing, D.-H. Shin, M. Zhang, and J. Zhang, "Personalized location privacy in mobile networks: A social group utility approach," in *Proc. IEEE Conf. Comput. Commun.*, 2015, pp. 1008–1016.
[28] Y. Xiao and L. Xiong, "Protecting locations with differential privacy under temporal correlations," in *Proc. 22nd ACM SIGSAC Conf. Comput. Commun. Secur.*, 2015, pp. 1298–1309.
[29] X. Yi, R. Paulet, E. Bertino, and V. Varadharajan, "Practical approximate k nearest neighbor queries with location and query privacy," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 6, pp. 1546–1559, Jun. 2016.
[30] K. He, J. Chen, R. Du, Q. Wu, G. Xue, and X. Zhang, "DeyPoS: Deduplicatable dynamic proof of storage for multi-user environments," *IEEE Trans. Comput.*, vol. 65, no. 12, pp. 3631–3645, Dec. 2016.
[31] Q. Wang, M. He, M. Du, S. S. M. Chow, R. W. F. Lai, and Q. Zou, "Searchable encryption over feature-rich data," *IEEE Trans. Depend. Secure Comput.*, to be published, doi: 10.1109/TDSC.2016.2593444.
[32] J. Chen, K. He, Q. Yuan, G. Xue, R. Du, and L. Wang, "Batch identification game model for invalid signatures in wireless mobile networks," *IEEE Trans. Mobile Comput.*, vol. 16, no. 6, pp. 1530–1543, Jun. 2017.
[33] C. Gentry, A. Sahai, and B. Waters, "Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based," in *Proc. Adv. Cryptology*, 2013, pp. 75–92.
[34] I. Damgård and M. Jurik, "A generalisation, a simplification and some applications of paillier's probabilistic public-key system," in *Proc. 4th Int. Workshop Practice Theory Public Key Cryptography: Public Key Cryptography*, 2001, pp. 119–136.
[35] B. Gedik and L. Liu, "Protecting location privacy with personalized k-anonymity: Architecture and algorithms," *IEEE Trans. Mobile Comput.*, vol. 7, no. 1, pp. 1–18, Jan. 2008.
[36] G. Sun, D. Liao, H. Li, H. Yu, and V. Chang, "L2P2: A location-label based approach for privacy preserving in LBS," *Future Gener. Comput. Syst.*, vol. 74, pp. 375–384, 2017.
[37] X. Huang, J. K. Liu, S. Tang, Y. Xiang, K. Liang, L. Xu, and J. Zhou, "Cost-effective authentic and anonymous data sharing with forward security," *IEEE Trans. Comput.*, vol. 64, no. 4, pp. 971–983, Apr. 2015.
[38] J. Kang, R. Yu, X. Huang, M. Jonsson, H. Bogucka, S. Gjessing, and Y. Zhang, "Location privacy attacks and defenses in cloud-enabled internet of vehicles," *IEEE Wireless Commun.*, vol. 23, no. 5, pp. 52–59, Oct. 2016.
[39] 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 Trans. Netw.*, vol. 25, no. 3, pp. 1703–1716, Jun. 2017.
[40] D. Yang, X. Fang, and G. Xue, "Truthful incentive mechanisms for k-anonymity location privacy," in *Proc. INFOCOM*, pp. 2994–3002, 2013.
[41] H. Kido, Y. Yanagisawa, and T. Satoh, "Protection of location privacy using dummies for location-based services," in *Proc. 21st Int. Conf. Data Eng. Workshops*, 2005, pp. 1248–1248.
[42] C. A. Ardagna, M. Cremonini, E. Damiani, S. D. C. d. Vimercati, and P. Samarati, "Location privacy protection through obfuscation-based techniques," in *Proc. 21st Annu. IFIP WG 11.3 Working Conf. Data Appl. Secur.*, 2007, pp. 47–60.
[43] B. Hoh, M. Gruteser, H. Xiong, and A. Alrabady, "Preserving privacy in gps traces via uncertainty-aware path cloaking," in *Proc. 14th ACM Conf. Comput. Commun. Secur.*, 2007, pp. 161–171.
[44] J. Meyerowitz and R. Roy Choudhury, "Hiding stars with fireworks: Location privacy through camouflage," in *Proc. 15th Annu. Int. Conf. Mobile Comput. Netw.*, 2009, pp. 345–356.
[45] D. Song and K. Park, "A privacy-preserving location-based system for continuous spatial queries," *Mobile Inf. Syst.*, vol. 2016, 2016, Art. no. 6182769.
[46] M. Li, S. Salinas, A. Thapa, and P. Li, "n-CD: A geometric approach to preserving location privacy in location-based services," in *Proc. INFOCOM*, 2013, pp. 3012–3020.
[47] Q. Ma, S. Zhang, T. Zhu, K. Liu, L. Zhang, W. He, and Y. Liu, "PLP: Protecting location privacy against correlation analyze attack in crowdsensing," *IEEE Trans. Mobile Comput.*, vol. 16, no. 9, pp. 2588–2598, Sep. 2017.
[48] Y. Gong, C. Zhang, Y. Fang, and J. Sun, "Protecting location privacy for task allocation in ad hoc mobile cloud computing," *IEEE Trans. Emerg. Topics Comput.*, vol. 6, no. 1, pp. 110–121, Jan.–Mar. 2018.
[49] L. Wang, D. Yang, X. Han, T. Wang, D. Zhang, and X. Ma, "Location privacy-preserving task allocation for mobile crowdsensing with differential geo-obfuscation," in *Proc. 26th Int. Conf. World Wide Web*, 2017, pp. 627–636.
[50] H. Zhu, R. Lu, C. Huang, L. Chen, and H. Li, "An efficient privacy-preserving location-based services query scheme in outsourced cloud," *IEEE Trans. Veh. Technol.*, vol. 65, no. 9, pp. 7729–7739, Sep. 2016.



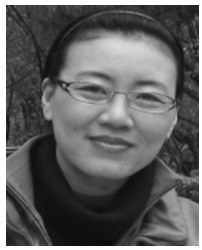
Jing Chen received the PhD degree in computer science from the Huazhong University of Science and Technology, Wuhan. He worked as an associate professor from 2010. His research interests in computer science are in the areas of network security and cloud security. He is the chief investigator of several projects in network and system security, funded by the National Natural Science Foundation of China (NSFC). He has published more than 60 research papers in many international journals and conferences, such as the *IEEE Transactions on Parallel and Distributed System*, the *International Journal of Parallel and Distributed System*, INFOCOM, SECON, TrustCom, and NSS. He acts as a reviewer for many journals and conferences, such as the *IEEE Transactions on Wireless Communication*, the *IEEE Transactions on Industrial Informatics*, *Computer Communications*, and GLOBCOM.



Kun He received the PhD degree in computer science from Wuhan University. His research interests include cryptography, network security, mobile computing, and cloud computing. He has published research papers in the *IEEE Transactions on Parallel and Distributed System*, the *International Journal of Communication Systems*, *Security and Communication Networks*, and IEEE TRUSTCOM.

Quan Yuan is an assistant professor in the Department of Math and Computer Science, University of Texas-Permian Basin, Texas. His research interests include mobile computing, routing protocols, peer-to-peer computing, parallel and distributed systems, and computer networks. He has published more than 30 research papers in many international journals and conferences, such as the *IEEE Transactions on Parallel and Distributed Systems*, INFOCOM, MobiHoc, SECON, and TrustCom.

Min Chen has been a full professor in the School of Computer Science and Technology, Huazhong University of Science and Technology (HUST) since February 2012. He is the director of the Embedded and Pervasive Computing (EPIC) Lab, HUST. He is a chair of the IEEE Computer Society (CS) Special Technical Communities (STC) on Big Data. He was an assistant professor in the School of Computer Science and Engineering, Seoul National University (SNU). He worked as a post-doctoral fellow in the Department of Electrical and Computer Engineering, at the University of British Columbia (UBC) for three years. Before joining UBC, he was a post-doctoral fellow with SNU for one and half years. He received the Best Paper Award from QShine 2008, IEEE ICC 2012, ICST IndustrialIoT 2016, and IEEE IWCMC 2016. He serves as an editor or associate editor for *Information Sciences*, *Information Fusion*, and *IEEE Access*, etc. He is a guest editor for the *IEEE Network*, the *IEEE Wireless Communications*, *IEEE Transactions Service Computing*, etc. He is a co-chair of the IEEE ICC 2012-Communications Theory Symposium, and co-chair of the IEEE ICC 2013-Wireless Networks Symposium. He is a general co-chair for IEEE CIT-2012, Tridentcom 2014, Mobimedia 2015, and Tridentcom 2017. He was a keynote speaker for CyberC 2012, Mobiquitous 2012, Cloudcomp 2015, IndustrialIoT 2016, Tridentcom 2017, and the 7th Brainstorming Workshop on 5G Wireless. His research interests include cyber physical systems, IoT sensing, 5G networks, mobile cloud computing, SDN, healthcare big data, medica cloud privacy and security, body area networks, emotion communications and robotics, etc. He has more than 300 paper publications, including 200+ SCI papers, more than 80 IEEE Trans./Journal papers, 16 ISI highly cited papers, and eight hot papers. He has published four books: *OPNET IoT Simulation* (2015), *Big Data Inspiration* (2015), *5G Software Defined Networks* (2016), and *Introduction to Cognitive Computing* (2017) with HUST Press, and *Big Data: Related Technologies*, *Challenges and Future Prospects* (2014) and *Cloud Based 5G Wireless Networks* (2016) with Springer, *Cognitive Computing and Deep Learning* (2018) with China Machine Press, and *Big Data Analytics for Cloud/IoT and Cognitive Computing* (2017) with Wiley. His Google Scholars Citations has reached 11,450+ with an h-index of 53. His top paper was cited 1150+ times. He has been an IEEE Senior Member since 2009. He received the IEEE Communications Society Fred W. Ellersick Prize in 2017. His research focuses on cyber physical systems, IoT sensing, 5G networks, mobile cloud computing, SDN, healthcare big data, medica cloud privacy and security, body area networks, emotion communications and robotics, etc.



Ruiying Du received the BS, MS, and PhD degrees in computer science from Wuhan University, Wuhan, China, in 1987, 1994, and 2008, respectively. She is a professor in the Computer School, Wuhan University. Her research interests include network security, wireless network, cloud computing, and mobile computing. She has published more than 80 research papers in many international journals and conferences, such as the *IEEE Transactions on Parallel and Distributed System*, the *International Journal of Parallel and Distributed System*, INFOCOM, SECON, TrustCom, and NSS.

Yang Xiang received the PhD degree in computer science from Deakin University, Australia. He is currently a full professor in the School of Information Technology, Deakin University. His research interests include network and system security, distributed systems, and networking. He has published more than 130 research papers in many international journals and conferences, such as the *IEEE Transactions on Computers* and the *IEEE Transactions on Information Security and Forensics*. He has been a PC member for more than 60 international conferences in networking and security. He serves as the an associate editor of the *IEEE Transactions on Computers and Security and Communication Networks* (Wiley).