An Efficient Privacy-Preserving Location Based Services Query Scheme in Outsourced Cloud

Hui Zhu, *Member, IEEE,* Rongxing Lu*, *Member, IEEE,* Cheng Huang, Le Chen, and Hui Li *Member, IEEE*

***Abstract*—With the pervasiveness of location-aware mobile electronic devices and the advances of wireless communication, location based services (LBS), which can help people enjoy convenient life, has attracted considerable interests recently. However, the privacy issues of LBS are still challenging today. Aiming at the challenges, in this paper, we present a new efficient and privacy-preserving LBS query scheme in outsourced cloud, named EPQ, for pervasive smart phones. In EPQ scheme, the LBS provider's data are first outsourced to the cloud server in an encrypted manner, and later a registered user can get accurate LBS query results without divulging his/her location information to the LBS provider and the cloud server. Specifically, based on an improved homomorphic encryption technique over composite order group, a special spatial range query algorithm SRQC over ciphertext is proposed, with which EPQ achieves privacy preservation of user's query and confidentiality of LBS data in the outsourced cloud server. Through detailed security analysis, we show that EPQ can resist various known security threats. In addition, we also implement EPQ over a smart phone and three workstations with a real LBS dataset, and extensive simulation results further demonstrate that the proposed EPQ scheme is highly efficient at smart phone side and can be implemented effectively in the cloud server.**

***Index Terms*—Location based services, security, privacy preserving, outsourced cloud.**

I. Introduction

In today's ubiquitous computing, the location based services (LBS), such as a map, friend or restaurant finder, can help people enjoy convenient life, and has recently attracted considerable interests [1]–[3]. In fact, due to the boom of smart phones and wireless communications, LBS has been prevalent in almost all social and business domains, and it is estimated that more than a billion people have enjoyed location based services in 2013. For example, when a tourist is out of his/her comfort zone, locating some places of interest, such as hotels, restaurants, hospitals, schools and shops, is necessary. As shown in Fig.1, through the accurate location-based services provided in smart phone, a user can query and search some of these facilities with distance (e.g., 500 meters) nearby, and it is convenient for the user to walk to these places.

Although LBS can benefit people by providing convenient lifestyle, the flourish of LBS system still hinges upon how we can fully understand and manage its challenges including information security and privacy preservation, especially in users' location privacy aspect [4]. For instance, by collecting and analyzing a user's current location and request content, the LBS provider can easily get plenty of sensitive information of the user [5]. To clearly illustrate the challenges in LBS system, we consider the following application scenario, where a user may request LBS to find hospitals by exposing his/her location to the LBS provider. However, the LBS provider can infer user's healthy status according to user's information. Therefore, how to design a secure and efficient privacy-preserving query scheme for LBS system has attracted considerable interest recently, and many research efforts have been dedicated to design privacy-preserving schemes for LBS.

By masking the user's real identity, the pseudo-identity technique can solve the privacy problem in access authentication [6]. However, for the LBS system, the location information of users can easily lead to the reveal of their real identity [7], e.g., frequently querying for e-coupons of the stores near a residential area with a pseudo-identity will reveal the user's home address. By blurring user's exact location point into a cloaked area, location-fuzzy (such as *k*-anonymity) is an important model in the area of privacy preservation [8]–[10]. In general, a trusted third party (TTP), which stores the original locations of LBS data and query, is required in location-fuzzy to ensure the probability of user be identified is very low. However, since TTP knows too much sensitive information, it easily becomes the target of attacks. Although some improved schemes (such as, private information retrieval [6], [11]) were proposed to improve the security of TTP, most of them bring much communication or computation cost on the user side, which leads to much energy consuming on the mobile devices. In addition, the LBS system with location-fuzzy cannot provide LBS accurately when preserving user's location information.

To achieve low computational cost and convenient data process, LBS providers often outsource their data to a cloud server, which will handle users' LBS queries by counting on its great computation power. In general, since the data is its sensitive and private asset, the LBS provider wants to keep the LBS data secret from the cloud server, and thus all data are encrypted before being outsourced to the cloud server; meanwhile, the users also want to keep their locations secret from the LBS provider and cloud server, thus the LBS query requests should also be encrypted. To achieve accurate access control and fuzzy search on encrypted LBS data, attribute-based encryption (ABE) [12] can be employed in the LBS systems, such as anonymous ciphertext-policy attribute-based encryption [13]. However, most of the existing ABE schemes require massive resource-consuming computa-

H. Zhu, C. Huang and H. Li are with the State Key Laboratory of Integrated Services Networks, Xidian University, Xi'an, 710071, China. E-mail: zhuhui@xidian.edu.cn; xdhuangcheng@gmail.com; lihui@mail.xidian.edu.cn.

R. Lu and L. Chen are with School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore. 639798. E-mail: rxlu@ntu.edu.sg; knnnn.evacl@gmail.com.

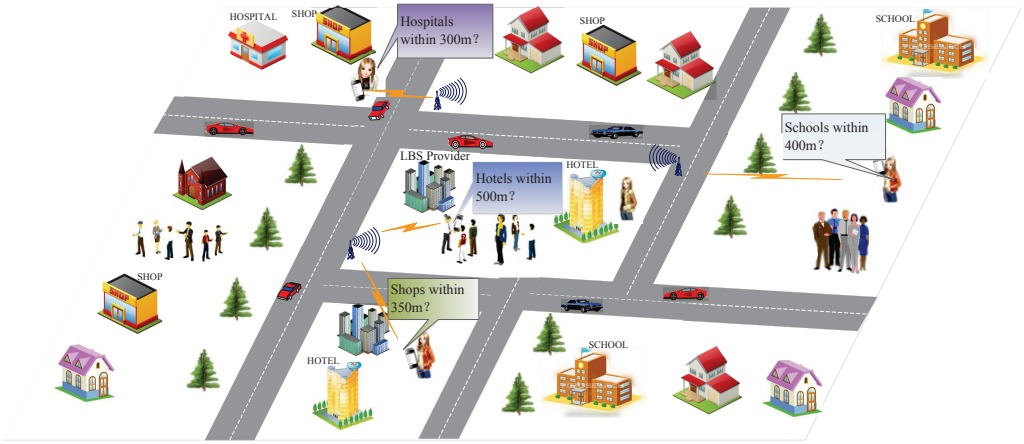


Fig. 1. The conceptual architecture of location based services.

tion, which makes it not quite suitable for resource-constraint mobile devices.

In this paper, aiming at the above challenges, we propose a new secure and efficient privacy-preserving location based services query scheme in outsourced cloud, called EPQ, for smart phone. The proposed EPQ scheme is characterized by employing an improved homomorphic encryption technique over composite order group to protect users' location privacy and the confidentiality of the LBS data with low overhead in computation and communication. Specifically, the main contributions of this paper are threefold.

- First, we propose EPQ, a secure and efficient privacy-preserving location based services query scheme for smart phone. With EPQ, the user can keep his/her query information secret from the LBS provider and cloud server, meanwhile, the LBS provider can also keep the LBS data secret from the cloud server. Since the user's queries and the LBS data are computed in the cloud server during the process of LBS query, to minimize the privacy disclosure due to the analysis of cloud server, EPQ introduces an improved homomorphic encryption technique over composite order group to only allow the registered user to query and obtain the desirable LBS data.
- Second, the EPQ scheme provides efficient and accurate LBS for smart phone. The overhead of communication is very low, since the process of LBS query is executed between the LBS user and cloud server without directly involving LBS provider. Moreover, different from other time-consuming homomorphic encryption techniques, we construct a special spatial range query algorithm SRQC over composite order group. By using Pollard's lambda method [14], the proposed SRQC algorithm can provide accurate LBS and achieve better efficiency in terms of computation overhead in smart phone and cloud server.
- Third, to evaluate the effectiveness of the proposed EPQ scheme in LBS system, we also develop a demo application built in Java, and test through a smart phone and three workstations with a real LBS dataset. Extensive simulation results demonstrate that the proposed EPQ can provide an efficient privacy-preserving LBS query.

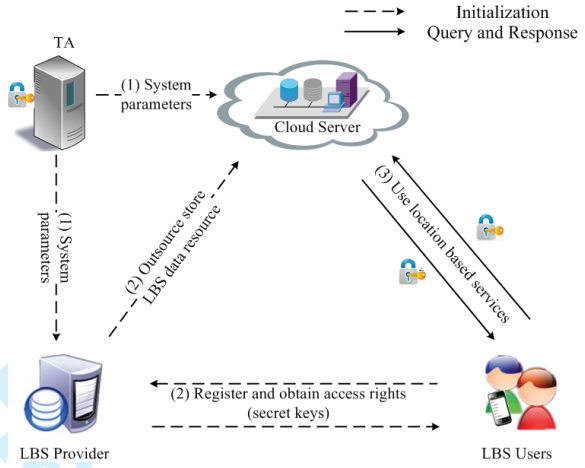


Fig. 2. System model under consideration.

The remainder of this paper is organized as follows. In Section II, we formalize the system model and security model, and identify our design goal. In Section III, we recall the bilinear pairings and 2DNF cryptosystem as the preliminaries. Then, we present our EPQ scheme in Section IV, followed by the security analysis and performance evaluation in Section V and Section VI, respectively. We also review some related works in Section VII. Finally, we draw our conclusions in Section VIII.

II. Models and Design Goal

In this section, we formalize the system model, security requirements, and identify our design goal.

A. System Model

In our system model, we mainly focus on how the LBS provider offers accurate and efficient LBS to users based on the requested privacy-preserving location data. Specifically, the system consists of four parts: *trusted authority* (TA), *LBS provider*, *LBS user*, and *cloud server*, as shown in Fig. 2.

- The trusted authority is a trusted third party which bootstraps the system initialization by generating and sending

system parameters to the LBS provider and cloud server, respectively.

- The LBS provider (i.e., the data owner) has abundant and accurate LBS data, which are corresponding to variant resources in real world, and can provide LBS to the registered users. With the advance of cloud computing, the LBS providers tend to outsource their LBS data to the cloud server. Therefore, the LBS provider mainly performs two functions: outsourcing data to the cloud server and providing users' registration system. With the process of outsourcing to the cloud server, the LBS provider will perform some encryption operations to guarantee the LBS data's confidentiality, while the users' register system is used to authenticate users in the cloud server.
- The cloud server (i.e., the data storage or process server) stores more than millions of encrypted LBS data items from the LBS provider, and provides query services for the LBS users. The cloud server also performs two functions: authentication and search over encrypted data. The authentication component is used to check users' identity, while the spatial range query in ciphertext component is used to search the encrypted LBS data items with the encryption of user's query. Furthermore, although the cloud server is featured with high performance in computation and storage, since thousands of users will query the LBS data at the same time, the efficiencies of computation and communication are still challenging.
- With the actual location information, the LBS user (i.e., the data user) who is registered in the LBS provider can query the accurate LBS data items which are outsourced to the cloud server. To guarantee the privacy of user's query, the LBS user will perform some encryption operations during the process of LBS query. Moreover, in order to lower energy cost, the encryption efficiency in mobile devices is very important.

B. Security requirements

The confidentiality of LBS data and the privacy of user's query are crucial for the success of secure LBS application. In our security model, we consider the cloud server, LBS provider, and LBS user are *honest-but-curious*. Specifically, the LBS provider provides the LBS data accurately, but it is curious to the LBS user's query location; the cloud server honestly executes the operations to search the required LBS data for user, but it also tries to analyse the encryption of stored LBS data to obtain the LBS data in real world, and guess the query location of user according to the user's request; moreover, the LBS user may try to access LBS data without registering or to access LBS data which are out of his/her accessing privileges. Therefore, in order to guarantee the confidentiality of LBS data in the cloud server and the privacy of user's query, the following security requirements should be satisfied in a secure LBS system.

- *Privacy*. On one hand, keeping the LBS provider's sensitive data assets secret from the cloud server, i.e., even if the cloud server stores all the data from the LBS provider and queries from the LBS users, it cannot identify any LBS data item. On the other hand, protecting the LBS user's query location from the cloud server, LBS provider and other users, i.e., even if the cloud server obtains all queries from the LBS users and all the responses to the LBS users, it cannot identify the LBS users' query location accurately. At this circumstance, the LBS data resource and user's location information can guarantee the privacy-preserving requirements. In addition, the privacy requirement also includes the cloud server's responses, i.e., only the legal LBS user can decrypt them. Note that, in our current model, we do not consider any two parties collude to disclose the third party's privacy. That is, the collusion attack on privacy is beyond the scope of this work, and will be discussed in future research.
- *Authentication*. Authenticating an encrypted LBS query that is really sent by a legal LBS user and has not been altered during the transmission, i.e., if an illegal LBS user forges an LBS query, this malicious operation should be detected. Besides, only the correct queries can be received by the cloud server. Meanwhile, the responses from the cloud server should also be authenticated so that the LBS users can receive authentic and reliable query results.

C. Design Goal

Under the aforementioned system model and security requirements, our design goal is to develop an efficient privacy-preserving location based services query scheme in outsourced cloud for smart phone, which will provide privacy-preserving LBS data and user's location information with accurate LBS for users. Specifically, the following three objects should be achieved.

- *The security requirements should be guaranteed*. If the LBS system does not consider the security, the LBS users' query locations and the sensitive data assets of LBS provider could be disclosed. Then, the LBS system cannot step into its flourish. Thus, the proposed scheme should achieve the confidentiality and authentication simultaneously.
- *LBS with high accuracy should be guaranteed*. The user experience is one of the most critical aspects of the LBS system, and it is important that the precision of LBS cannot be lowered while protecting user's privacy. Therefore, the proposed scheme should also provide highly precise and reliable LBS.
- *The effectiveness of proposed scheme in computation and communication should be achieved*. Although the performance of smart phone is continuously improved today, its battery is still limited. Moreover, although the cloud server is considered to have great computation power, the improvement in computational efficiency can also reduce the energy consumption and the cost of purchasing hardware. As a result, the proposed scheme should consider the effectiveness in terms of computation and communication to reduce the power consumption of smart phone and cloud server.

III. Preliminaries

In this section, we recall the bilinear pairing technique and review the 2DNF Cryptosystem [15], which will serve as the basis of our proposed EPQ scheme.

A. Bilinear Pairing of Composite Order

Let $\mathbb{G}$, $\mathbb{G}_T$ be two (multiplicative) cyclic groups of the same composite order n, where $n = q_1 q_2$ is the product of two primes q_1, q_2, and g be a generator of $\mathbb{G}$. Suppose $\mathbb{G}$ and $\mathbb{G}_T$ are equipped with a pairing, and a non-degenerated and efficiently computable bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ has the following properties: i) Bilinearity. For all $u, v \in \mathbb{G}$ and $a, b \in \mathbb{Z}_n$, we have $e(u^a, v^b) = e(u, v)^{ab}$; ii) Non-degeneracy. $e(g, g) \neq 1_{\mathbb{G}_T}$; and iii) Computability. $e(u, v)$ can be computed efficiently for all $u, v \in \mathbb{G}$.

B. 2DNF Cryptosystem

The 2DNF cryptosystem [15] can achieve the homomorphic properties, which is similar to the Paillier [16] and Okamoto-Uchiyama [17] encryption schemes. Concretely, the 2DNF Cryptosystem is comprised of three parts: key generation, encryption and decryption.

- Key Generation ($Gen(\kappa)$). Given a security parameter $\kappa \in \mathbb{Z}^+$, two κ-bit primes numbers q_1, q_2 are first chosen, and $n = q_1 \cdot q_2 \in \mathbb{Z}$ is computed. Generate a bilinear group $\mathbb{G}$ of order n, and let g, u be two generators of $\mathbb{G}$. Then, $h = u^{q_2}$ is calculated as a random generator of the subgroup of $\mathbb{G}$ with order q_1. Finally, private key $SK = q_1$ and public key $PK = (n, \mathbb{G}, \mathbb{G}_T, e, g, h)$ are outputted.
- Encryption. Assume that the message space consists of integers in the set $\{0, 1, \cdots, T\}$ with $T < q_2$, then, to encrypt a message m with public key PK, we select a random r from $\{0, 1, \cdots, n-1\}$ and the ciphertext can be calculated by $C = g^m \cdot h^r \in \mathbb{G}$.
- Decryption. To decrypt a ciphertext C with private key $SK = q_1$, be aware of $C^{q_1} = (g^m \cdot h^r)^{q_1} = (g^{q_1})^m$, let $\hat{g} = g^{q_1}$. To achieve the corresponding message m, it suffices to compute the discrete logarithm of $C^{q_1} = \hat{g}^m$ base $\hat{g}$, since $0 \leq m \leq T$ only takes the expected time $\hat{O}(\sqrt{T})$ using Pollard's lambda method.

Note that the decryption time in system would be the polynomial time in the size of the message space T_s. Hence, it is very suitable for encrypting short messages.

IV. Proposed EPQ Scheme

In this section, we present our efficient and privacy-preserving location based services query EPQ scheme in outsourced cloud for smart phone, which mainly consists of three phases: *system initialization, cloud server data creation, privacy-preserving location based services*. For easier expression, we give the description of notations to be used in EPQ in TABLE II.

TABLE I
Definition of Notations in the Scheme

Notation	Definition
N_s	the serial number of LBS data items in LBS provider
T_j	the kinds of resouces
D_s	the description of the resource s
$H()$	th secure cryptographic hash functions
d	the search range of query, with meter as unit
(x_s, y_s)	the location coordinate of resource s, with meter as unit
(x_{s_0}, y_{s_0})	the disturbed location coordinate of resource s
(x_c, y_c)	the location coordinate of user, with meter as unit
(x_0, y_0)	the disturbed location coordinate of user
$n = q_1 \cdot q_2$	q_1, q_2 are two big primes
$\mathbb{G}, \mathbb{G}_T$	the bilinear groups with order n
g, e	the parameters of bilinear groups
PB	$PB = e(g, g)^{q_1}$
SB	$SB = g^{q_1}$
$E()$	symmetric encryption algorithm, e.g., AES
k	the secret key for $E()$
$(l_{s_1}, l_{s_2}, l_{s_3}, l_{s_4})$	the search index related to the location of resource s
(rq_1, rq_2, rq_3, rq_4)	the key components related to location of user
TRL	the temporary resource list
EDS	the evaluation dataset

A. System Initialization

For a single-authority LBS system under consideration, we assume the trusted authority TA will bootstrap the whole system. Specifically, TA first chooses a security parameter κ to obtain $(\mathbb{G}, \mathbb{G}_T, q_1, q_2, e, g, h, n = q_1 \cdot q_2)$ by running $Gen(\kappa)$, computes and keeps two secret bases, $SB = g^{q_1}$ and $PB = e(g, g)^{q_1}$. Then, TA chooses a random number $s_{TA} \in \mathbb{Z}_n^*$ as its private key SK_{TA}, computes its public key $PK_{TA} = g^{SK_{TA}}$. In addition, TA also chooses a secure symmetric encryption algorithm $E()$, i.e., AES and a secure cryptographic hash function $H()$, where $H : \{0,1\}^* \to \mathbb{Z}_n^*$ in the system. Finally, TA keeps the tuple $< q_1, SK_{TA} >$ as the master key secretly and publishes the system parameters $< n, \mathbb{G}, \mathbb{G}_T, e, g, h, PK_{TA}, E(), H() >$.

The LBS provider chooses random number $s_{lbs} \in \mathbb{Z}_n^*$ as its private key SK_{LBS}, and computes and the corresponding public key $PK_{LBS} = g^{SK_{LBS}}$. Similarly, the cloud server chooses its public and private key pair as $(PK_{CS} = g^{SK_{CS}}, SK_{CS})$, and each LBS user U_i also chooses a random number $s_i \in \mathbb{Z}_n^*$ as its private key SK_{U_i} and computes its public key $PK_{U_i} = g^{SK_{U_i}}$. When the LBS provider registers itself to the TA, TA sends $< SB, PB >$ back to the LBS provider in a secure channel, and goes offline or suffers slowdowns to against the single point of attack. Then, the LBS provider chooses a random number $k \in \mathbb{Z}_n^*$ as the secret key of $E()$ to encrypt its LBS resources. As a registered user of the LBS provider, each U_i is authorized with $< SB, PB, PK_{CS}, E(), H(), k >$, which will be utilized for retrieving LBS resources in a privacy-preserving way later. Note that, the cloud server is not given (SB, PB); and the LBS provider will publish a list of registered users to the cloud server, so that the latter can authenticate these users.

B. Cloud Server Data Creation

In general, the LBS provider has plenty of LBS resources, and most of resources' information (such as number, type, coordinate, description, etc.) stored in the LBS provider are plaintext in the format of $< N_s, T_j, (x_s, y_s), D >$ as shown below.

$$\begin{vmatrix} < 1, & T_1, & (x_1, y_1), & D_1 > \\ < 2, & T_2, & (x_2, y_2), & D_2 > \\ < 3, & T_1, & (x_3, y_3), & D_3 > \\ & \cdots & & \\ < s, & T_j, & (x_s, y_s), & D_s > \end{vmatrix}$$

Before being uploaded to the cloud server, each LBS data item N_s in the LBS provider, for security reason, is processed as follows.

- The LBS provider firstly obtains location information (x_s, y_s) from the data item N_s, and computes $x_{s_0} = x_s + H(k)$ and $y_{s_0} = y_s + H(k)$ to increase the sample space of location information, which can resist the exhaustive attack.
- The LBS provider chooses two random numbers $r_{s_1}, r_{s_2} \in \mathbb{Z}_n^*$, and computes the encrypted coordinate index $(l_{s_1}, l_{s_2}, l_{s_3}, l_{s_4})$, which can be implicitly formed by the following formulas.

$$\begin{cases} l_{s_1} = PB^{{x_{s_0}}^2}; & l_{s_2} = PB^{{y_{s_0}}^2} \\ l_{s_3} = g^{x_{s_0}} \cdot h^{r_{s_1}}; & l_{s_4} = g^{y_{s_0}} \cdot h^{r_{s_2}} \end{cases}$$

- The LBS provider computes the encrypted location data $E_s = E_k(x_s \| y_s \| D)$ with a secure symmetric encryption algorithm $E()$ and a secret key k.
- The LBS provider outsources the encrypted data item $< ID_{LBS} \| T_j \| l_{s_1} \| l_{s_2} \| l_{s_3} \| l_{s_4} \| E_s >$ to the cloud server.

Finally, in the cloud server, all LBS data items (such as number, affiliation, type, coordinate index, description in ciphertext, etc.) are stored in the form of $< N_s, ID_{LBS}, T_j, (l_{s_1}, l_{s_2}, l_{s_3}, l_{s_4}), E_s >$ as follows.

$$\begin{vmatrix} < 1, & ID_{LBS}, & T_1, & (l_{1_1}, l_{1_2}, l_{1_3}, l_{1_4}), & E_1 > \\ < 2, & ID_{LBS}, & T_2, & (l_{2_1}, l_{2_2}, l_{2_3}, l_{2_4}), & E_2 > \\ < 3, & ID_{LBS}, & T_3, & (l_{3_1}, l_{3_2}, l_{3_3}, l_{3_4}), & E_3 > \\ & & \cdots & & \\ < s, & ID_{LBS}, & T_j, & (l_{s_1}, l_{s_2}, l_{s_3}, l_{s_4}), & E_s > \end{vmatrix}$$

In addition, the LBS provider also computes $ED_i = H(PB^i)$, where $0 \le i \le 10,000,000,000$; and processes the collection of ED_i in a meaningfully ascending order (for example, data values from the smallest to the largest) to construct the evaluation dataset $EDS = \{ED_0, ED_1, \cdots, ED_i, \cdots, ED_{10,000,000,000}\}$; Finally, the dataset EDS will be uploaded to the cloud server. The function of EDS is to enable the cloud server to perform privacy-preserving query without privacy disclosure, the details will be described later.

C. Privacy-Preserving Location Based Services

1) LBS Query Generation: After registering in the LBS provider, U_i can securely send his/her query request to the cloud server and avoid exposing his/her actual location by the following procedure.

- U_i firstly obtains his/her location from his/her smart phone, denoting (x_c, y_c) as the location information, and computes $x_0 = x_c + H(k)$, $y_0 = y_c + H(k)$, where the key k is only known by the LBS provider and registered users.
- U_i decides the search range d from location (x_c, y_c) which he/she wants to query, and computes the encrypted query (rq_1, rq_2, rq_3, rq_4), which can be implicitly formed as

$$\begin{cases} rq_1 = PB^{{x_0}^2 - d^2}; & rq_2 = PB^{{y_0}^2} \\ rq_3 = SB^{(2 \cdot x_0)}; & rq_4 = SB^{(2 \cdot y_0)} \end{cases}$$

- U_i encrypts the LBS query request $E_{LQR} = E_{PK_{CS}}(rq_1 \| rq_2 \| rq_3 \| rq_4)$ by the cloud server's public key PK_{CS}.
- U_i makes a signature Sig_i as $Sig_i = (H(ID_{LBS} \| E_{LQR} \| U_i \| TS))^{s_i}$ by using his/her private key $SK_{U_i} = s_i$, where TS is the current time stamp, which can resist the potential replay attack.
- U_i sends the encrypted LBS query request $< ID_{LBS} \| E_{LQR} \| U_i \| TS \| Sig_i >$ to the cloud server.

2) Privacy-Preserving Search and Response: Upon receiving U_i's encrypted LBS query request $< ID_{LBS} \| E_{LQR} \| U_i \| TS \| Sig_i >$, the cloud server provides the query service by the following procedure.

- The cloud server first checks the time stamp TS and the signature Sig_i to verify its validity, i.e., verify whether $e(g, Sig_i) = e(PK_{U_i}, H(ID_{LBS} \| E_{LQR} \| U_i \| TS))$. If it does hold, the signature is accepted, since $e(g, Sig_i) = e(g, (H(ID_{LBS} \| E_{LQR} \| U_i \| TS))^{x_i} = e(PK_{U_i}, H(ID_{LBS} \| E_{LQR} \| U_i \| TS))$.
- After the validity checking, the cloud server decrypts E_{LQR} by its secret key SK_{CS} to obtain rq_1, rq_2, rq_3 and rq_4, and executes the spatial range query algorithm SRQC as follows.
  - For each data item N_s stored in the cloud server, the cloud server firstly computes the search criteria T_s, where T_s is implicitly formed by

$$\begin{aligned} T_s &= \frac{e(l_{s_3}, rq_3) \cdot e(l_{s_4}, rq_4)}{rq_1 \cdot rq_2 \cdot l_{s_1} \cdot l_{s_2}} \\ &= \frac{e(g^{x_{s_0}} \cdot h^{r_{s_1}}, SB^{2x_0}) \cdot e(g^{y_{s_0}} \cdot h^{r_{s_2}}, SB^{2y_0})}{PB^{{x_0}^2 - d^2} \cdot PB^{{y_0}^2} \cdot PB^{{x_{s_0}}^2} \cdot PB^{{y_{s_0}}^2}} \\ &= \frac{e(g^{x_{s_0}} \cdot h^{r_{s_1}}, g^{q_1 \cdot 2x_0}) \cdot e(g^{y_{s_0}} \cdot h^{r_{s_2}}, g^{q_1 \cdot 2y_0})}{PB^{{x_0}^2 - d^2} \cdot PB^{{y_0}^2} \cdot PB^{{x_{s_0}}^2} \cdot PB^{{y_{s_0}}^2}} \\ &= \frac{e(g^{x_{s_0} \cdot q_1}, g^{2x_0}) \cdot e(g^{y_{s_0} \cdot q_1}, g^{2y_0})}{PB^{{x_0}^2 - d^2} \cdot PB^{{y_0}^2} \cdot PB^{{x_{s_0}}^2} \cdot PB^{{y_{s_0}}^2}} \\ &= \frac{e(g, g)^{q_1 \cdot 2x_{s_0} x_0} \cdot e(g, g)^{q_1 \cdot 2y_0 y_{s_0}}}{PB^{{x_0}^2 - d^2} \cdot PB^{{y_0}^2} \cdot PB^{{x_{s_0}}^2} \cdot PB^{{y_{s_0}}^2}} \\ &= PB^{d^2 - ((x_0 - x_{s_0})^2 + (y_0 - y_{s_0})^2)} \\ &= PB^{d^2 - ((x_c - x_s)^2 + (y_c - y_s)^2)}. \end{aligned}$$

  - The cloud server computes $ST_s = H(T_s)$ and searches ST_s in the evaluation dataset EDS by running the binary search algorithm. If ST_s is not found in EDS, the current encrypted data item N_s

does not meet the user's requirements; otherwise, the cloud server records the encrypted description E_s from current encrypted data item N_s in the temporary resource list (TRL).

– After traversing through all the LBS data items in the cloud server one by one, a TRL with w items will be stored as $< 1, E_i >, < 2, E_j >, < 3, E_k >, \cdots, < w, E_l >$.

Correctness of the SRQC algorithm. Taking a close look at the exponential of search criteria $T_s = PB^{d^2-((x_c-x_s)^2+(y_c-y_s)^2)}$, we know $(x_c-x_s)^2+(y_c-y_s)^2$ is the square of the distance between user's query location and resource's location, which is less than 20,000 kilometers on earth, i.e. $0 \leq (x_c-x_s)^2+(y_c-y_s)^2 \leq 4\times10^{14} < 2^{49}$. In addition, the search range d is also usually less than 100 kilometers, i.e. $0 \leq d^2 \leq 10,000,000,000$. Therefore, if the resource N_s meets the user's requirements, i.e., $0 \leq d^2-((x_c-x_s)^2+(y_c-y_s)^2) \leq d^2 \leq 10,000,000,000$ and the corresponding ST_s must be in EDS; otherwise, $-2^{49} \leq d^2-((x_c-x_s)^2+(y_c-y_s)^2) < 0$, and $T_s = PB^{d^2-((x_c-x_s)^2+(y_c-y_s)^2)} = PB^{q_2+d^2-((x_c-x_s)^2+(y_c-y_s)^2)}$, then the corresponding ST_s will not be in EDS, since $(q_2+d^2-((x_c-x_s)^2+(y_c-y_s)^2)) \gg 10,000,000,000$. As a result, the correctness follows.

- The cloud server encrypts the TRL with the secure symmetric encryption algorithm $E()$ and rq_1, i.e., $E_{rq_1}(TRL)$, and makes a signature $Sig_{cs} = H(E_{rq_1}(TRL)||ID_{cs}||TS)^{SK_{cs}}$ by its private key SK_{cs}. Finally, the cloud server sends $< E_{rq_1}(TRL)||ID_{cs}||TS||Sig_{cs} >$ to U_i.

3) Query Result Reading: Upon receiving $< E_{rq_1}(TRL)||ID_{cs}||TS||Sig_{cs} >$ from the cloud server, the U_i first verifies the validity by checking $e(g, Sig_{cs}) = e(PK_{cs}, H(E_{rq_1}(TRL)||ID_{cs}||TS))$, and then decrypts the items in TLS with the secret key rq_1 and k to read the encrypted query results.

V. Security Analysis

In this section, we analyze the security properties of the proposed EPQ scheme. Specifically, following the security requirements discussed earlier, our analysis will focus on how the proposed EPQ scheme can achieve the LBS data confidentiality, the user's query location privacy, and source authentication of the query request and response.

- *The user query location is privacy-preserving in the proposed EPQ scheme.* In the proposed EPQ scheme, the user's LBS query request is encrypted in the form of (rq_1, rq_2, rq_3, rq_4), where $rq_1 = PB^{d^2-x_0^2}$, $rq_2 = PB^{y_0^2}$, $rq_3 = SB^{(2\cdot x_0)}$, and $rq_4 = SB^{(2\cdot y_0)}$, before being sent to the cloud server. And to avoid the exhaustive attack against rq_1, rq_2, rq_3 and rq_4 by Pollard's lambda method, the sample space of user's query location is increased by computing $x_0 = x_c + H(k)$ and $y_0 = y_c + H(k)$. Since k, PB and SB are only known by the LBS provider and the registered users, and the collusion attack is not considered in current security model, the cloud server cannot obtain the user's actual query location (x_c, y_c) according to his/her query requests. Specifically, the encrypted user's queries and the encrypted LBS data are computed in the cloud server to obtain the encrypted result, which will be sent back to the smart phone, and the cloud server also cannot obtain any useful information of LBS resources, even in the continuous search queries environment. Meanwhile, the cloud server still provides the accurate service to the registered users by the proposed position contrast algorithm. In particular, according to the exponential of search criteria $T_s = PB^{d^2-((x_c-x_s)^2+(y_c-y_s)^2)}$, the cloud server can easily find all the eligible LBS data items by traversing through the search index of LBS data items, but it cannot obtain any useful information of user's location. Moreover, both the registered user and cloud server do not interact with the LBS provider when the LBS query is processing, and the user's LBS query request is encrypted by the cloud server's public key PK_{CS} before being sent to the cloud server, so the LBS provider cannot obtain the actual location of user. In addition, before being sent to U_i, the LBS query result is encrypted by rq_1, which is only known by U_i and the cloud server, and other registered users and the LBS provider cannot get U_i's query result. Therefore, from the above aspects, the LBS user's query location is privacy-preserving in the proposed EPQ scheme.
- *The proposed EPQ scheme can achieve confidential LBS data.* Specifically, the cloud sever cannot obtain the actual location information of the resource, although it can get all the outsourced data items and users' query information. In the proposed EPQ scheme, before the LBS provider publishes its data items to the cloud server, each item's coordinate and description have been encrypted with a secure symmetrical encryption algorithm in the form of $E_{is} = E_k(x_s||y_s||D_s)$, where the encryption key k is kept secret from the cloud server. And the corresponding search index $(l_{s_1}, l_{s_2}, l_{s_3}, l_{s_4})$, which can be implicitly expressed as $l_{s_1} = PB^{x_{s_0}^2}$, $l_{s_2} = PB^{y_{s_0}^2}$, $l_{s_3} = g^{x_{s_0}} \cdot h^{r_{s_1}}$, and $l_{s_4} = g^{y_{s_0}} \cdot h^{r_{s_2}}$, has been computed to achieve efficient search over encrypted data items. Moreover, to avoid the exhaustive attack against $l_{s_1}, l_{s_2}, l_{s_3}$ and l_{s_4} by Pollard's lambda method, the sample space of source's actual location is also increased by computing $x_{s_0} = x_s + H(k)$ and $y_{s_0} = y_s + H(k)$. In addition, to avoid the guessing attacks for PB in the evaluation dataset EDS, the relationship between PB and ED_i is hidden by a secure hash function $H()$. Therefore, from the above four aspects, the cloud server cannot obtain the resource's actual location (x_s, y_s) according to the outsourced LBS data items.
- *The authentication of the LBS query request and response are achieved in the proposed EPQ scheme.* In the proposed EPQ scheme, each registered user's request and the response of cloud server are signed by BLS short signature [21]. Since the BLS short signature is provably secure under the CDH problem in the random oracle model, the source authentication can be guaranteed. Moreover, for any unregistered user, since he/she doesn't have the

secret key k and g^{q_1}, he/she also cannot submit valid query request to the cloud server. As a result, the query request from the unregistered user and the response from the mendacious cloud server in the LBS system can be detected in the proposed EPQ scheme.

From the above analysis, we can conclude that the proposed EPQ scheme is secure and privacy-preserving, and can achieve the desirable security goal in the security model under consideration.

VI. Performance Evaluation

In this section, we evaluate the performance of our proposed EPQ scheme in terms of the computational complexity of the cloud server, LBS provider and LBS user with a real LBS dataset. In order to measure the integrated performance of EPQ in real environment, we also implement EPQ on a smart phone and three workstations in a wireless network by using a custom simulator built in Java.

A. Computation Complexity

As the core components of the proposed EPQ scheme (i.e., EPQ is an improved SRQC scheme with authenticated and secure channel), the spatial range query algorithm SRQC is effective and can solve the time-consuming issue at cloud server and LBS user. In specific, when a user U_i generates an encrypted query (rq_1, rq_2, rq_3, rq_4), it requires 4 exponentiation operations in $\mathbb{Z}_{q_2}$. After receiving the query from U_i, the cloud server firstly computes the search criteria T_s for each data item N_s stored in the cloud server, which includes $2N$ pairing operations and $4N$ multiplication operations for checking all N resource items. After obtaining the response from cloud server, U_i need decrypt the items with symmetrical encryption algorithm, which is considered negligible compared to exponentiation and pairing operations. Moreover, the LBS provider need $2N$ pairing operations, $2N$ multiplication operations and $6N$ exponentiation operations in the *Cloud Server Data Creation* phase. Denote the computational costs of an exponentiation operation in $\mathbb{Z}_{q_2}$, a multiplication operation in $\mathbb{G}$ /$\mathbb{G}_t$, and a pairing operation by C_p, C_m, and C_e, respectively. Then, totally for the user, the cloud server and the LBS provider, the computational costs will be $4*C_e$, $2N*C_p+4N*C_m$, and $2N*C_p+2N*C_m+6N*C_e$ in the proposed SRQC algorithm.

Different from other time-consuming homomorphic encryption techniques, the proposed special spatial range query algorithm SRQC uses Pollard's lambda method over composite order group. It can provide accurate LBS and largely reduce the encryption times for the smart phone and cloud server. In the following, for the comparison with SRQC, we selected a traditional attribute-based encryption approach (denoted by FINE) [4], where the information exchange via an authenticated and secure channel. Denote the query ranges of x-axis and y-axis by Δx and Δy , and let l be the number of the resources which satisfy the request. Therefore, the computational costs of the user, the cloud server and the LBS provider will be $l*C_p+(2+l)*C_m+2l*C_e$, $2N\Delta x\Delta y*C_p+5N\Delta x\Delta y*C_m+(4N\Delta x\Delta y+2l)*C_e$, and $2N*C_p + 3N*C_m + N(8N+6)*C_e$, respectively.

We present the computation complexity comparison of the proposed SRQC and FINE in TABLE II, and it is obvious that our proposed SRQC scheme can achieve privacy-preserving location based services with low computation complexity in the cloud server, LBS provider, and smart phone.

B. Simulation and Evaluation

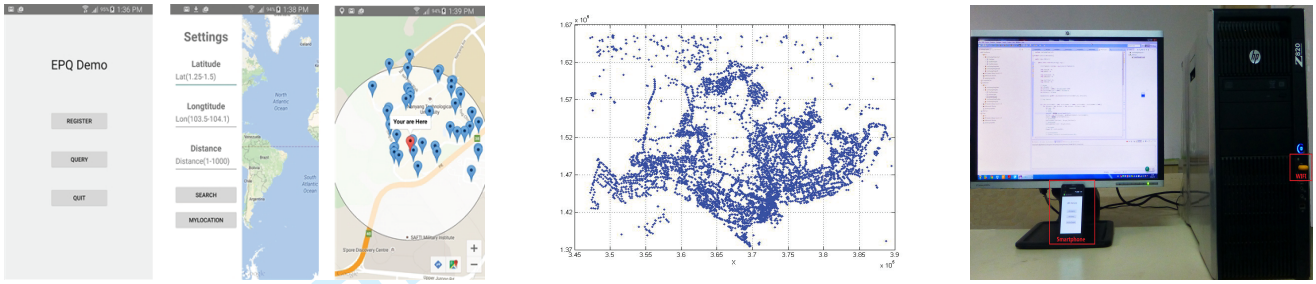
1) Evaluation Environment: Firstly, a smart phone with 1.4 GHz quad-core processor, 2GB RAM, Android 4.0, and three workstations with 2.0 GHz 6-core processor, 64GB RAM, Windows 7, are chosen to evaluate the LBS user, LBS provider, cloud server and TA respectively. Specifically, based on our proposed EPQ scheme, an LBS application built by JAVA, named EPQ.apk, is installed in the smart phone, the simulators of LBS provider, cloud server and TA are deployed in three workstations, and the information of resources and corresponding encryption information are stored in the LBS provider and cloud server. As shown in Fig. 3.(a), users can register in the LBS provider, query coordinate in the cloud server, and display result in the smart phone by EPQ.apk. Consider road networks are important spatial constraints in location privacy protection [18], as shown in Fig. 3.(b), the LBS resources' geographic coordinates in LBS provider are collected from the *open street map in Singapore* [19], which has 10,836 LBS resource items. In addition, to reduce the computation cost of cloud server in system implementation, the resources dataset can be divided into plurality of basic areas, and each basic area has sufficient resources to hide the query coordination. For example, the dataset in Fig. 3.(b) can be divided into $I \times J$ areas, and each area has a unique number *RAN* which is stored in the LBS provider and sent to the cloud server as an attribute of resources in the area. In specific, if an area number is (i, j),$1 \le i \le I, 1 \le j \le J$, the coordinate range of this area can be determined, and upon receiving an LBS query from the LBS user, the cloud server will perform the following steps.

- *STEP 1*. The cloud server selects a resource from each area randomly to construct a temporary dataset *TD*, and performs the *Privacy-Preserving Search* operation over dataset *TD*. If no resource satisfies the query requirements, does STEP 1 again; otherwise, the satisfied resource and its neighbour areas' *RAN* will be recorded in a related area list (*RAL*). For example, if a resource in area (i, j) satisfies the query requirements, then area (i, j), $(i-1, j-1)$, $(i-1, j)$, $(i-1, j+1)$, $(i, j-1)$, $(i, j+1)$, $(i+1, j-1)$, $(i+1, j)$ and $(i+1, j+1)$ will be recorded in *RAL*.
- *STEP 2*. The cloud server performs the *Privacy-Preserving Search* operations over the resources whose *RAN* are recorded in *RAL*, and sends all satisfied resources to LBS user.

2) The cloud server: In the proposed EPQ scheme, when the cloud server receives a query request from the user, it will compute the search criteria T_s using bilinear pairing over composite order group for each resource in the query region,

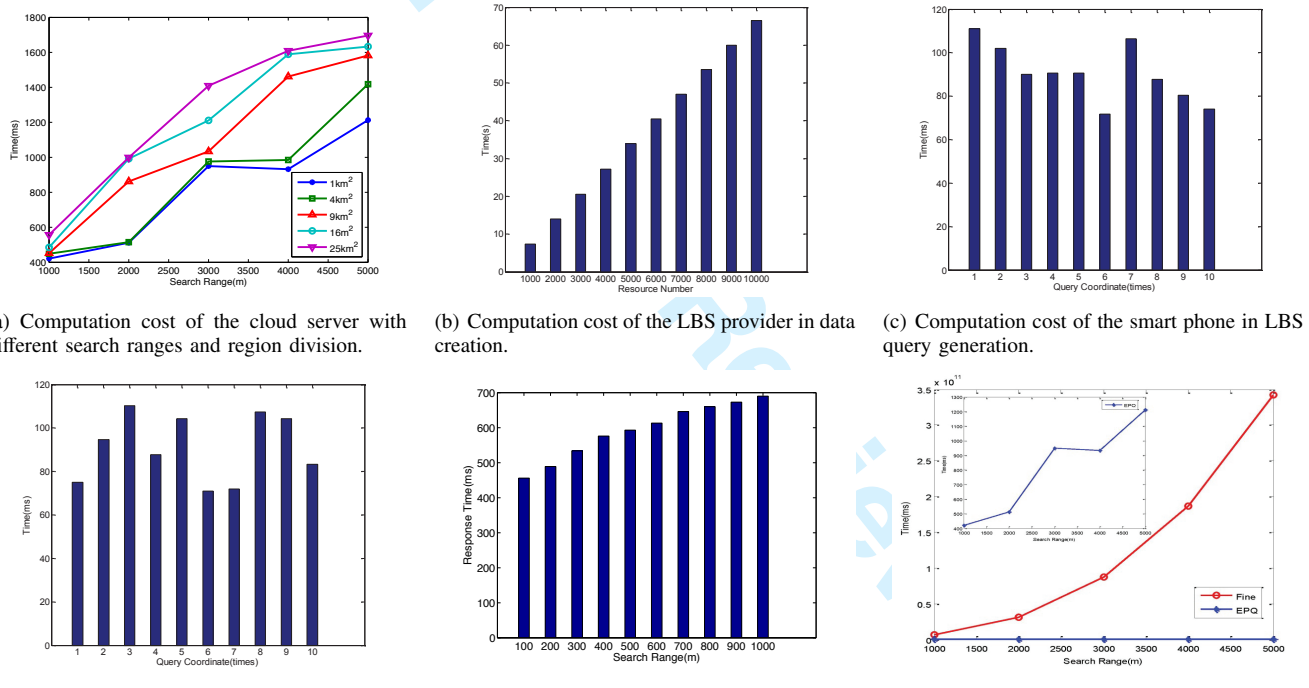
TABLE II
COMPARISON OF COMPUTATION COMPLEXITY

	Proposed SRQC in EPQ	FINE [4]
User	$4*C_e$	$l*C_p+(2+l)*C_m+2l*C_e$
Cloud server	$2N*C_p+4N*C_m$	$2N\Delta x\Delta y*C_p+5N\Delta x\Delta y*C_m+(4N\Delta x\Delta y+2l)*C_e$
LBS provider	$2N*C_p+2N*C_m+6N*C_e$	$2N*C_p+3N*C_m+N(8N+6)*C_e$



(a) User interface of EPQ.apk. (b) Coordinations of Singapore's resources. (c) Prototype for LBS query.

Fig. 3. Evaluation environment of EPQ.



(a) Computation cost of the cloud server with different search ranges and region division.

(b) Computation cost of the LBS provider in data creation.

(c) Computation cost of the smart phone in LBS query generation.

(d) Computation cost of the smart phone in query result reading.

(e) Query response time in real environment.

(f) Average running time in Cloud vs Fine.

Fig. 4. Computation complexity of EPQ.

which is the mainly computation overhead of cloud server. i.e., the number of resources in the query region and the search range d of user's query request may impact the computation complexity in the cloud server. Therefore, different region division methods and search ranges are chosen to illustrate the computation cost of cloud server. Specifically, the open street map of Singapore is divided into many equal basic areas with 1, 4, 9, 16, and 25 square kilometers respectively, and the search ranges of requests are selected from 1,000 to 5,000 meters. For each search range and region division method, we select 10 different geographic coordinates of Singapore to query and calculate the average value of the 10 computation overheads in the cloud server. As shown in Fig. 4.(a), the computation cost of cloud server rises slowly with the increase of the search range and the square meters of region, and the computation costs in could server are less than 1 second for various region division methods when the search range $d \leq 2000$ meters. Note that the computation costs in could server are also between 1 and 1.8 seconds for searching all the open street map resources of Singapore, even through $d = 5000$ meters. It is obvious that the computation cost in cloud server is acceptable.

3) The LBS provider: In our proposed EPQ scheme, all the compute operations of LBS provider are in the *Cloud Server Data Creation* phase, and the factor impacting the computation cost of *system initialization* is the number of resources in the LBS provider. Therefore, different numbers of resources in the open street map of Singapore are chosen to illustrate the computation cost of LBS provider. As shown in Fig. 4.(b), the numbers of resources are selected from 1,000 to 10,000. It is obvious that increasing the number of resources linearly increases the computation cost of LBS provider. The reason is that, when the LBS provider publishes the dataset of resources' geographic coordinates to the cloud server, each geographic coordinate will be operated to obtain the corresponding $(l_{s_1}, l_{s_2}, l_{s_3}, l_{s_4})$ and E_s, which will cost more time with the increase of resources' number.

4) The LBS user: The query response time of the LBS user (i.e. smart phone) is an important result illustrating our proposed EPQ scheme, and the compute operations in the smart phone are in the phases *LBS Query Generation* and *Query Result Reading*. Therefore, different LBS queries and search ranges are chosen to illustrate the computation cost of smart phone. To observe the computation cost of smart phone, 10 query coordinates are selected randomly from the open street map of Singapore, and 10 different search ranges for each query coordinate are selected from 1,000 to 5,000 meters; then, we execute 100 times with different query coordinates and search ranges, and calculate the average computation cost in different phases for each query coordinate. Specifically, Fig.4.(c) shows that the average computation cost of smart phone in the phase *LBS Query Generation* for 10 different query coordinates, and Fig.4.(d) shows that the corresponding computation cost in the phase *Query Result Reading*. It is obvious that the total computation cost in smart phone is less than 200 milliseconds, and need only once communication. For the comparison with the performance of user in EPQ, we select and implement a traditional private information retrieve approach (denoted by TKDE) [11], where the information search is over plaintext. The experimental results in smart phone indicate that users in TKDE need about 2 seconds and 3 times communications for sending a complete information query to server.

5) Integrated performance in real environment: In order to evaluate the integrated performance of our proposed EPQ scheme, the EPQ scheme is deployed in a real environment with the resources of open street map. Specifically, 10,836 resources in Fig. 3.(b) are selected, and the information of resources and corresponding encryption information are stored in the LBS provider and cloud server, respectively. In addition, the smart phone and cloud server are connected through a 802.11g WLAN as Fig. 3.(c), and when users input the query coordination and search range by EPQ.apk, the smart phone will send a query request to the cloud server and get the response through WLAN. Therefore, we run 1,000 times to evaluate the performance of EPQ with different configurations (10 different search ranges (from 100 to 1,000 meters) for 100 different query coordinations of Singapore) in a real environment. In Fig. 4.(e), we plot the average response time of EPQ varying with different search ranges d. The results show that the entire overheads for once whole privacy-preserving LBS query are consistent with the results in the simulation environment and all the query response time are less than 0.7 seconds in the real environment. For the comparison with EPQ, we also plot the computation overheads of EPQ and FINE [4] varying with different search ranges d in the open street map database of Singapore in Fig. 4.(f). From the figure, we can see that by increasing d, the computation overhead of the FINE protocol increase hugely, which is much higher than that of our proposed EPQ protocol. It can be obviously shown that the EPQ scheme largely reduces the computation complexity for both the user and the cloud server.

From the above analysis, the proposed EPQ scheme is indeed efficient in terms of computation and communication cost, which is suitable for the smart phone and cloud server in the real location based services environment.

VII. Related Works

The study of privacy-preserving location based services has gained great interest from the research community recently, and we briefly review some of them closely related to ours.

Gruteser [20] firstly introduced the Location k-Anonymity Model, which ensures that a user cannot be identified with a probability at least $1/k$. Although the adversary does not infer the actual location in general, the location information will be leaked if k users' locations were in the same place or in a sensitive region (e.g., hospital). To achieve the privacy-preserving management of location information, Gedik et al. [21] presented that a trusted third party (TTP) is required to achieve the anonymization of user's actual location. And Khoshgozaran et al. [22] proposed an approach where a TTP converts the original location of LBS data and query into another space. The TTP should maintain the spatial relationship between the LBS data and query; otherwise, the LBS user cannot obtain the accurate query result. In addition, Zhong et al. [23] introduced a distributed approach that integrates nicely with existing infrastructures for location-based services, based on homomorphic encryption. Chow et al. [24] introduced a casper system, where a TTP named location anonymizer is used to blur user's exact location point into a specified size of cloaked area containing at least $k-1$ other users. However, since the TTP knows too much sensitive information of users, it would easily be the target of attacks.

Kido et al. [25] introduced the "dummy" locations, and the user should contain many random other locations in his/her LBS query to hide his/her actual location. However, the communication overhead and computation cost in smart phone will be increased, since choosing, sending and receiving plenty of random data related to the fake locations. Therefore, by using private information retrieve (PIR), Paulet et al. [11] presented a scheme to keep the location secret without TTP. The user can obtain the record from a database without revealing which record he/she is interested in through PIR. However, all the information of resource data are operated in plaintext and known by the cloud server in PIR schemes, which does not satisfy the outsourced environment. Jung et al. [26] proposed a privacy-preserving LBS system without

TTP by using ABE. However, their solution brings heavy computation burden to the user side due to the computation of pairings [27]. To reduce the computation cost on the user side, Shao et al. [4] introduced a fine-grained privacy-preserving LBS framework, named FINE, which provided accurate query result and location privacy, and the computation cost in smart phone is acceptable. However, the cost in the cloud server is huge and hard to implement in the server. In addition, consider road networks are important spatial constraints in location privacy protection, Wang et al. [18] present a general model for privacy-aware mobile services.

Different from the above works, our proposed EPQ scheme aims at the efficiency and privacy issues, and based on an improved homomorphic encryption technique over composite order group, we develop an efficient privacy-preserving LBS system in outsourced cloud for smart phone. In particular, the proposed EPQ scheme can easily implement in the smart phone and cloud server, and the processing of LBS query is just needed in the cloud server. The computation costs in both smart phone and cloud server are acceptable.

VIII. Conclusions

In this paper, we have proposed a secure, efficient, and privacy-preserving location based services query scheme in outsourced cloud, called EPQ, for smart phone. Based on an improved homomorphic encryption technique over composite order group, the proposed EPQ can achieve location privacy preservation and confidentiality of LBS data. Specifically, for an LBS query request from a registered user, the LBS query execution is directly performed over ciphertext on the cloud server without decryption, and the result of LBS query can only be decrypted by the registered user. Thus, the user can get accurate LBS query result without divulging his/her location information. Detailed security analysis shows its security strength and privacy-preserving ability, and extensive experiments are conducted to demonstrate its efficiency. In the future work, we will take into consideration of the collusion attack and lowering the trust level in the cloud server from honest-but-curious model.

Availability

The implementation of the proposed EPQ scheme and relevant information can be available at `http://ste.xidian.edu.cn/zhuhui/EPQ/`.

References

[1] E.-C. Lu, V. S. Tseng, and P. S. Yu, "Mining cluster-based temporal mobile sequential patterns in location-based service environments," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 23, no. 6, pp. 914–927, 2011.

[2] H. Zhu, T. Liu, G. Wei, and H. Li, "Ppas: privacy protection authentication scheme for vanet," *Cluster computing*, vol. 16, no. 4, pp. 873–886, 2013.

[3] R. Lu, X. Lin, and X. Shen, "Spoc: A secure and privacy-preserving opportunistic computing framework for mobile-healthcare emergency," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 24, no. 3, pp. 614–624, 2013.

[4] J. Shao, R. Lu, and X. Lin, "Fine: A fine-grained privacy-preserving location-based service framework for mobile devices," in *INFOCOM, 2014 Proceedings IEEE*. IEEE, 2014, pp. 1452–1461.

[5] M. Li, H. Zhu, Z. Gao, S. Chen, L. Yu, S. Hu, and K. Ren, "All your location are belong to us: Breaking mobile social networks for automated user location tracking," in *Proceedings of the 15th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, ser. MobiHoc '14. New York, NY, USA: ACM, 2014, pp. 43–52.

[6] G. Ghinita, P. Kalnis, A. Khoshgozaran, C. Shahabi, and K.-L. Tan, "Private queries in location based services: anonymizers are not necessary," in *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*. ACM, 2008, pp. 121–132.

[7] H. Hu, Q. Chen, and J. Xu, "Verdict: Privacy-preserving authentication of range queries in location-based services," in *Data Engineering (ICDE), 2013 IEEE 29th International Conference on*. IEEE, 2013, pp. 1312–1315.

[8] Q. Wang, C. Xu, and M. Sun, "Multi-dimensional k-anonymity based on mapping for protecting privacy," *Journal of Software*, vol. 6, no. 10, pp. 1937–1944, 2011.

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

[10] L. Sweeney, "Achieving k-anonymity privacy protection using generalization and suppression," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, no. 5, pp. 571–588, 2002.

[11] M. Russell Paulet, G. Kaosar, X. Yi, and E. Bertino, "Privacy-preserving and content-protecting location based queries," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 26, no. 5, 2014.

[12] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proceedings of the 13th ACM conference on Computer and communications security*. ACM, 2006, pp. 89–98.

[13] Y. S. Rao and R. Dutta, "Recipient anonymous ciphertext-policy attribute based encryption," in *Information Systems Security*. Springer, 2013, pp. 329–344.

[14] A. J. Menezes, P. C. Van Oorschot, and S. A. Vanstone, *Handbook of applied cryptography*. CRC press, 2010.

[15] D. Boneh, E.-J. Goh, and K. Nissim, "Evaluating 2-dnf formulas on ciphertexts," in *Theory of cryptography*. Springer, 2005, pp. 325–341.

[16] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *Advances in cryptologyłEUROCRYPT99*. Springer, 1999, pp. 223–238.

[17] T. Okamoto and S. Uchiyama, "A new public-key cryptosystem as secure as factoring," in *Advances in CryptologyłEUROCRYPT'98*. Springer, 1998, pp. 308–318.

[18] T. Wang and L. Liu, "Privacy-aware mobile services over road networks," *Proc. VLDB Endow.*, vol. 2, no. 1, pp. 1042–1053, Aug. 2009.

[19] O. Foundation, "Openstreetmap," http://www.openstreetmap.org/#map=10/1.3375/103.9732, 2015.

[20] M. Gruteser and D. Grunwald, "Anonymous usage of location-based services through spatial and temporal cloaking," in *Proceedings of the 1st International Conference on Mobile Systems, Applications and Services*, ser. MobiSys '03. New York, NY, USA: ACM, 2003, pp. 31–42.

[21] B. Gedik and L. Liu, "Location privacy in mobile systems: A personalized anonymization model," in *Distributed Computing Systems, 2005. ICDCS 2005. Proceedings. 25th IEEE International Conference on*. IEEE, 2005, pp. 620–629.

[22] A. Khoshgozaran and C. Shahabi, "Blind evaluation of nearest neighbor queries using space transformation to preserve location privacy," in *Advances in Spatial and Temporal Databases*. Springer, 2007, pp. 239–257.

[23] G. Zhong and U. Hengartner, "Toward a distributed k-anonymity protocol for location privacy," in *Proceedings of the 7th ACM workshop on Privacy in the electronic society*. ACM, 2008, pp. 33–38.

[24] C.-Y. Chow, M. F. Mokbel, and W. G. Aref, "Casper*: Query processing for location services without compromising privacy," *ACM Transactions on Database Systems (TODS)*, vol. 34, no. 4, p. 24, 2009.

[25] H. Kido, Y. Yanagisawa, and T. Satoh, "Protection of location privacy using dummies for location-based services," in *Data Engineering Workshops, 2005. 21st International Conference on*. IEEE, 2005, pp. 1248–1248.

[26] P. McDaniel and S. McLaughlin, "Security and privacy challenges in the smart grid," *IEEE Security and Privacy*, vol. 7, no. 3, pp. 75–77, 2009.

[27] Y. Zhu, D. Ma, D. Huang, and C. Hu, "Enabling secure location-based services in mobile cloud computing," in *Proceedings of the second ACM SIGCOMM workshop on Mobile cloud computing*. ACM, 2013, pp. 27–32.