FINE: A Fine-Grained Privacy-Preserving Location-based Service Framework for Mobile Devices

Jun Shao
Zhejiang Gongshang University
Hangzhou 310018, China
Email: chn.junshao@gmail.com

Rongxing Lu
Nanyang Technological University
Singapore 639798
Email: rxlu@ntu.edu.sg

Xiaodong Lin
University of Ontario Institute of Technology
Ontario, Canada
Email: xiaodong.lin@uoit.ca

***Abstract*—In this paper, we propose a fine-grained privacy-preserving location-based service (LBS) framework, called FINE, for mobile devices. It adopts the data-as-a-service (DaaS) model, where the LBS provider publishes its data to a third party (e.g., cloud server) who executes users' LBS queries. The proposed FINE framework employs a ciphertext-policy anonymous attribute-based encryption technique to achieve fine-grained access control, location privacy, confidentiality of the LBS data and its access policy, and accurate LBS query result while without involving any trusted third party. Moreover, the proposed FINE framework also integrates the transformation key and proxy re-encryption to migrate most of computation-intensive tasks from the LBS provider and users to the cloud server. This property keeps mobile devices away from massive resource-consuming operations. Extensive analysis shows that our proposed FINE framework is secure and highly efficient for mobile devices in terms of computation and communication cost.**

I. Introduction

Location-Based Service (LBS) has recently become popular in almost all social and business domains due to the boom of location-aware mobile electronic devices [1]. Some LBS examples [2] include location-based sales, e.g., "Download e-coupons of those stores within two miles of my current location", and location-based store finders, e.g., "Where is the nearest restaurant to my current location". While LBS provides convenience to people, it also threatens user privacy significantly, especially the location privacy. By using the location information, a privacy-curious LBS provider can do lots of things. For example, they can infer a user's health status from the information regarding how often s/he reveals her/his location close to a hospital when requesting LBS. Although the pseudo-identity technique can mask the user's identity, it cannot really solve the privacy problem in LBS [2]. The location information, if not protected, can easily lead to the true identity. For instance, frequently asking for downloading e-coupons of those stores near a user's house with a pseudo-identity will reveal the user's true identity, i.e., as a member of the household.

To address the challenge, many research efforts have recently been dedicated to design privacy-preserving techniques for LBS [3]–[8]. K-anonymity, which ensures that a user cannot be identified with a probability at least 1/k, is an important technique to achieve user location privacy in LBS. Usually, a trusted third party (TTP) is required to achieve k-anonymity property, i.e., the TTP transforms the original location of LBS data and query into another space [3], or blurs user's exact location point into a cloaked area [4]. However, since the TTP knows too much sensitive information of users, it would easily be the single target of attacks. To avoid the necessity of TTP, "dummy" locations [9], private information retrieval (PIR) [5], [6] and secret circular shift [7] techniques are introduced into privacy-preserving LBS systems. However, most of existing techniques bring much communication or computation cost on the user side, which leads to much energy consuming on the mobile devices. To the best of our knowledge, how to design a TTP-free, location-preserving LBS system suitable for mobile devices is still challenging.

On the other hand, LBS usually falls into the data-as-a-service (DaaS) model [1], where the LBS provider publishes its LBS data to a third party server (e.g., cloud server) who handles LBS queries. Meanwhile the LBS provider also wants to keep the LBS data secret from the cloud server, since the data is its sensitive and private asset. To achieve fine-grained access control on the LBS data, some LBS systems [8] apply attribute-based encryption (ABE) [10]. However, all existing ABE schemes require massive resource-consuming computation, which makes it not suitable for resource-constraint mobile devices. Although the access policy of the LBS data is also needed to protect from the viewpoint of LBS provider, there is no approach so far to this concern in the context of privacy-preserving LBS.

In this paper, aiming at addressing the above challenges, we propose a fine-grained privacy-preserving location-based service framework for mobile devices, called FINE. The proposed FINE framework is characterized by employing a ciphertext-policy anonymous attribute-based encryption (CP-AABE) technique to achieve fine-grained access control, location privacy, confidentiality of the LBS data and its access policy, and accurate LBS query result while without involving any trusted third party. However, ABE is not suitable for mobile device because of the huge computation cost [11], [12]. To solve the problem, the proposed FINE framework also

978-14799-3360-0/14/$31.00 ©2014 IEEE

TABLE I
COMPARISONS AMONG THE FINE FRAMEWORK AND OTHER LBS SYSTEMS

Property	FINE	KYS [9]	GKKST [5]	PKYB [6]	LLSW [7]	JL [8]
A[a]	✓	✓	✗	✗	✓	✗
B[b]	✓	✗	✓	✓	✓	✓
C[c]	✓	✗	✓	✓	✗	✓

[a]no massive resource-consuming operations
[b]accurate LBS query result
[c]constant cost on generation of LBS query

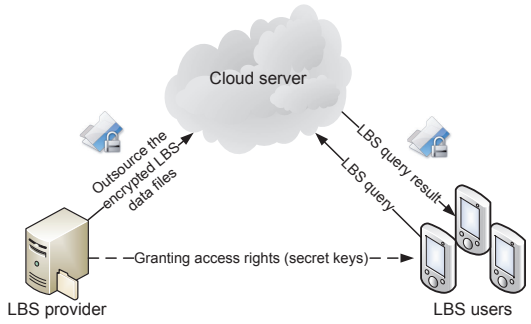

Fig. 1. Architecture of our proposed FINE framework

integrates the transformation key [11] and proxy re-encryption [13]–[15] to migrate most of computation intensive tasks from the LBS provider and users to the cloud server. To summarize, the main contributions of this paper are threefold.

- To the best of our knowledge, by uniquely integrating CP-AABE, transformation key and proxy re-encryption techniques, the FINE framework is the first work that simultaneously achieves location privacy, confidentiality of the LBS data and its access policy, TTP-freeness and fine-grained access control in the LBS system.
- The FINE framework is quite suitable for mobile devices, since users can receive LBS on a *minimum* computation and communication cost, where "minimum" means no massive resource-consuming operations, accurate LBS query result and constant cost on generation of LBS query. In Table I, we make the comparisons among the FINE framework and other TTP-free privacy-preserving LBS systems in terms of the above three properties, and the results show that the FINE framework outperforms others.
- The variant of CP-AABE may be of independent interest, since it allows range query with constant computation cost on query generation, no matter how many attributes exist in the system or how big the query range is.

The remainder of this paper is organized as follows. In Section II, we formalize the system model, security model, and identify our design goals. We present the details of our proposed FINE framework in Section III. In what follows, we give the security analysis and performance evaluation in Section IV and Section V, respectively. Section VI reviews the related works, and finally Section VII gives the conclusions of our paper.

II. MODELS AND DESIGN GOALS

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

A. System Model

Our model follows the data-as-a-service (DaaS) model [1] where the system is composed of the following parties: the LBS provider, many LBS users, and the cloud server as shown in Fig. 1. The LBS provider produces the LBS data (consisting of many LBS data files) that is outsourced to the cloud server. To achieve the confidentiality of LBS data, the data should be encrypted according to the corresponding access policy. The LBS user who has the access rights granted from the LBS provider can obtain the LBS data from the cloud server according to her/his LBS query. For simplicity, we denote $(L_x, L_y) \in \mathbb{Z}_q^2$ as the spatial coordinates of user's location, and its unit is measured in meter [8]. It is worth mentioning that the LBS user usually receives the LBS via her/his mobile device (i.e., smartphone, tablet and pad) with constrained resource, which demands that the decryption on the encrypted LBS data files should be as little as possible. While the cloud server is assumed to have abundant storage capacity and computation power, and it is always online. The LBS provider is assumed to be of enough computation resource, and it is online only when the LBS user grant or revocation operation happens. The key pair(s) of LBS provider will be updated, and the corresponding version number will be increased by one after every user revocation operation.

B. Security Model

We assume that the cloud server is *honest-but-curious* as that in [16]. That is to say, the cloud server will faithfully follow the proposed framework, but could launch passive attacks to get secret information as much as possible. Specifically, the cloud server will try to get the LBS data and its access policy, and the user's location information by colluding malicious users, but they won't modify the communication data between them and honest users or collude with the LBS provider. Users want to receive LBS while keeping their location information secret[1]. Meanwhile, they would try to access LBS data out of their access rights, or obtain the access policy of LBS data. To get these secret information, they may launch attacks independently or cooperatively. At last, the LBS provider is also curious on the user's location information, but it is only allowed to get the location information based on the data he obtains from communication with users and the cloud server.

C. Design Goals

Our design goal is to develop a fine-grained privacy-preserving location-based service framework for mobile devices. It has the following desirable properties.

[1]We assume that the origin of a packet in the networks can be successfully hidden by other techniques; otherwise, any adversary can obtain the location information just from the origin of the packet. However, these hiding techniques are out of this paper's scope.

- *Fine-grained access control:* The proposed framework should allow the LBS provider to specify an access policy directly on the encryption of LBS data instead of the access control list (ACL). In this case, the management of ACL could be replaced by the one of decryption keys, which is much easier to realize when the LBS data will be outsourced to the honest-but-curious cloud server.
- *Privacy-preserving protocol:* The proposed framework should achieve privacy requirements of the parties in the framework. In particular, i) the LBS data cannot be revealed to the one who has no access right; ii) the access policy of LBS data cannot be revealed to anyone, even if the legitimate user can only know the fact that s/he can access the LBS data; iii) users' location information cannot be revealed to anyone.
- *Accurate query result:* The proposed framework should guarantee that all received encrypted LBS data files *exactly* satisfy her/his LBS query (location information and access rights). This property allows users to save the bandwidth, subsequently saves energy for mobile devices.
- *Minimum operations on the user side:* The proposed framework should allow a user to get the LBS data only by minimum operations, including the generation of query and decryption on the encrypted LBS data files. These operations should not contain massive resource-consuming computation, and the amount should only be related to the number of satisfied LBS data files.

III. Proposed FINE Framework

In this section, we present our fine-grained privacy-preserving LBS framework (FINE), which consists of five parts: system initialization (SI), service data creation (SDC), user grant (UG), user revocation (UR), and location-based service (LBS). Before plugging into the details, we first review the preliminaries, including bilinear pairing [17], and ciphertext policy anonymous attribute-based encryption [10], [18], which will serve as the basis of the FINE framework.

A. Preliminaries

1) Bilinear Groups: Let $\mathbb{G}$ and $\mathbb{G}_T$ be two multiplicative cyclic groups of prime order q. These two groups are equipped with a non-degenerated and efficiently computable bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$, such that $e(g_1^a, g_2^b) = e(g_1, g_2)^{ab}$ for all $a, b \in \mathbb{Z}_q^*$ and any $g_1, g_2 \in \mathbb{G}$. We denote $\mathtt{BSetup}$ as an algorithm that, on the input of security parameter λ, outputs the parameters for a bilinear map as $(q, g_1, g_2, \mathbb{G}, \mathbb{G}_T, e)$, where $q \in \Theta(2^\lambda)$.

2) Ciphertext Policy Anonymous Attribute-Based Encryption (CP-AABE): CP-AABE [10], [18] is a kind of ciphertext policy attribute-based encryption (CP-ABE), which allows the data encryptor to embed the access policy $\mathbb{A}$ into the ciphertext during encryption, and each decryption is based on some attribute set $\mathbb{S}$. Hence the user is able to decrypt a ciphertext if and only if her/his attribute set $\mathbb{S}$ satisfies the corresponding access policy $\mathbb{A}$. Besides supporting data confidentiality like CP-ABE, CP-AABE supports policy-anonymity, i.e., even the legitimate decryptor cannot obtain the information about the access policy beyond the fact that s/he can decrypt the data. A CP-AABE scheme is composed of the following four algorithms like CP-ABE. In this paper, we only take the CP-AABE scheme in [18] as an example (named LRZW scheme), but other CP-AABE schemes can also be applied in the proposed FINE framework.

- Setup: Run $\mathtt{BSetup}(1^\lambda)$ to obtain $(q, g_1, g_2, \mathbb{G}, \mathbb{G}_T, e)$. Define a cryptographic hash function $H : \{0,1\}^* \to \mathbb{G}$. Suppose that there are n attributes ω_i for $i = 1, 2, \ldots, n$ in universe, each attribute has n_i multiple values v_{ij} for $j = 1, 2, \ldots, n_i$. Choose a random element α from $\mathbb{Z}_q^*$, and compute $T = e(g_1, g_2)^\alpha$. The key pair is
$$(pk, sk) = ((g_1, g_2, e, T, H), \alpha).$$
- KeyGen: To generate the secret key for a user with attribute set $\mathbb{S} = [S_1, S_2, \cdots, S_n] = [v_{1,k_1}, v_{1,k_2}, \cdots, v_{1,k_n}]$, where $1 \le k_i \le n_i$, choose random $\{\beta_i, r_i, r_i'\}_{1 \le i \le n-1}, r_n, r_n' \in \mathbb{Z}_q^*$, and compute $\beta_n = \alpha - \sum_{i=1}^{n-1} \beta_i \bmod q$. The secret key is
$$sk_{\mathbb{S}} = \{(d_{i0}, d_{i1}, d_{i0}', d_{i1}')\} = \{(g_2^{\beta_i} H(1||i||v_{i,k_i})^{r_i}, g_1^{r_i}, g_1^{\beta_i} H(0||i||v_{i,k_i})^{r_i'}, g_2^{r_i'})\}_{1 \le i \le n}$$
- Enc: To encrypt a message m from the message space under access policy $\mathbb{A} = [A_1, A_2, \cdots, A_n]$, choose a random value z from $\mathbb{Z}_q^*$ and compute $C_0 = mT^z$. For each $1 \le i \le n$ and $1 \le t_i \le n_i$,
  - if $v_{i,t_i} \in A_i$, choose z_{i,t_i} from $\mathbb{Z}_q^*$ and compute $(C_{i,t_i,0}, C_{i,t_i,1}, C'_{i,t_i,0}, C'_{i,t_i,1}) = (H(1||i||v_{i,t_i})^{z_{i,t_i}}, g_1^{z_{i,t_i}}, H(0||i||v_{i,t_i})^{z - z_{i,t_i}}, g_2^{z - z_{i,t_i}})$;
  - if $v_{i,t_i} \notin A_i$, choose z_{i,t_i}, z'_{i,t_i} from $\mathbb{Z}_q^*$ and compute $(C_{i,t_i,0}, C_{i,t_i,1}, C'_{i,t_i,0}, C'_{i,t_i,1}) = (H(1||i||v_{i,t_i})^{z_{i,t_i}}, g_1^{z_{i,t_i}}, H(0||i||v_{i,t_i})^{z'_{i,t_i}}, g_2^{z'_{i,t_i}})$.

  The output ciphertext is
$$C = (C_0, \{(C_{i,t_i,0}, C_{i,t_i,1}, C'_{i,t_i,0}, C'_{i,t_i,1}\})$$
  for $1 \le i \le n$ and $1 \le t_i \le n_i$.
- Dec: To decrypt the ciphertext $C = (C_0, \{(C_{i,t_i,0}, C_{i,t_i,1}, C'_{i,t_i,0}, C'_{i,t_i,1}\})$ without knowing the access policy $\mathbb{A}$, the user with the secret key $sk_{\mathbb{S}} = \{(d_{i0}, d_{i1}, d_{i0}', d_{i1}')\}$ on attribute set $\mathbb{S} = [S_1, S_2, \cdots, S_n] = [v_{1,k_1}, v_{1,k_2}, \cdots, v_{1,k_n}]$ computes $C' = \prod_{i=1}^{n} \frac{e(d_{i0}, C_{i,k_i,1}) e(d'_{i0}, C'_{i,k_i,1})}{e(d_{i1}, C_{i,k_i,0}) e(d'_{i1}, C'_{i,k_i,0})}$, and $m = C_0 / C'$.

In the proposed FINE framework, we additionally need the following variant of LRZW scheme.

- Setup: Almost the same as algorithm Setup, except that we need $(n+1)$-th attribute that is a dummy attribute. We denote the resultant key pair as $(\mathtt{pk}, \mathtt{sk})$.
- KeyGen: Almost the same as algorithm KeyGen, except that the key components related to $(n+1)$-th attribute: $(\mathtt{d}_{n+1}, \mathtt{d}'_{n+1}) = (g_2^{\bar{\beta}_{n+1}}, g_1^{\bar{\beta}_{n+1}})$, where $\bar{\alpha} =$

$\sum_{i=1}^{n+1} \bar{\beta}_i$, $\bar{\beta}_i$ $(i = 1, \cdots, n-1, n+1)$ are chosen randomly from $\mathbb{Z}_q^*$. Set $\mathtt{sk}_{\mathbb{S}} = (\mathtt{sk}_{\mathbb{S},1}, \mathtt{sk}_{\mathbb{S},2}) = (\{(\mathtt{d}_{i0}, \mathtt{d}_{i1}, \mathtt{d}'_{i0}, \mathtt{d}'_{i1})\}_{1 \le i \le n}, (\mathtt{d}_{n+1}, \mathtt{d}'_{n+1}))$.

- Enc: To "encrypt" $L_x, L_y \in \mathbb{Z}_q^*$ under access policy $\mathbb{A}$, it is almost the same as algorithm ENC, except that the ciphertext components related to $(n+1)$-th attribute and $\mathtt{C}_0, \mathtt{C}'_0$: Choose $\mathtt{z}_{n+1}$ from $\mathbb{Z}_q^*$ and compute $(\mathtt{C}_{n+1}, \mathtt{C}'_{n+1}) = (g_1^{\mathtt{z}_{n+1}}, g_2^{\mathtt{z}-\mathtt{z}_{n+1}})$, $\mathtt{C}_0 = \mathtt{T}^{\mathtt{z}}$, and $\mathtt{C}'_0 = e(g_1, H(1||g_1))^{\mathtt{z}_{n+1} \cdot L_x} \cdot e(g_2, H(2||g_2))^{(\mathtt{z}-\mathtt{z}_{n+1}) \cdot L_y}$. The output ciphertext is

$$\mathtt{C} = (\mathtt{C}_0, \mathtt{C}'_0, \mathtt{C}_{n+1}, \mathtt{C}'_{n+1}, \{(\mathtt{C}_{i,t_i,0}, \mathtt{C}_{i,t_i,1}, \mathtt{C}'_{i,t_i,0}, \mathtt{C}'_{i,t_i,1}\})$$

  for $1 \le i \le n$ and $1 \le t_i \le n_i$.
- Dec: We never use the secret key to obtain (L_x, L_y), but to check whether the ciphertext is encrypted according to (L_x, L_y). Without knowing the access policy $\mathbb{A}$, the user with the secret key $\mathtt{sk}_{\mathbb{S}}$ on attribute set $\mathbb{S}$ computes $\mathtt{C}' = e(\mathtt{d}_{n+1} H(1||g_1)^{L_x}, \mathtt{C}_{n+1}) \cdot e(\mathtt{d}'_{n+1} H(2||g_2)^{L_y}, \mathtt{C}'_{n+1}) \cdot \prod_{i=1}^{n} \frac{e(\mathtt{d}_{i0}, \mathtt{C}_{i,k_i,1}) e(\mathtt{d}'_{i0}, \mathtt{C}'_{i,k_i,1})}{e(\mathtt{d}_{i1}, \mathtt{C}_{i,k_i,0}) e(\mathtt{d}'_{i1}, \mathtt{C}'_{i,k_i,0})}$, and check whether $\mathtt{C}'/\mathtt{C}_0 = \mathtt{C}'_0$ holds. If it holds, then we can deduce that the ciphertext is encrypted according to (L_x, L_y) with a overwhelming probability; otherwise, it is not.

The correctness of the variant can be easily obtained from that of LRZW scheme.

B. Main Idea

CP-AABE plays an important role in our proposed FINE framework. In particular, by using CP-AABE, every LBS data file is encrypted according to its own access policy, and each user is granted secret keys according to her/his attribute set by the LBS provider. Hence, the fine-grained access control is obtained immediately. However, CP-AABE cannot fulfil FINE framework alone due to the following two problems.

- It will introduce heavy computation burden to the user when s/he performs decryption. It is because all existing CP-AABE schemes require massive resource-consuming operations, which is not suitable for resource-constraint mobile devices.
- It will also introduce heavy computation burden to the LBS provider when user revocation happens. In particular, the LBS provider should re-encrypt all the LBS data files related to the revoked user, and re-granted the access rights to remained legitimate users.

We utilize the transformation key [11] to solve the first problem. In particular, it allows a honest-but-curious proxy to transform CP-AABE ciphertexts into ElGamal [19] type ciphertexts. To decrypt the ElGamal type ciphertext, no pairing computation is needed. In the FINE framework, the cloud server acts as the proxy. Only if the LBS data file remains the same, the transformation is performed only once for the same user. For the second problem, we utilize the bidirectional proxy re-encryption (BPRE) [13] that allows a honest-but-curious proxy to translate ciphertexts among different public keys, while plaintexts are kept secret from the proxy. In the FINE framework, the cloud server always translates ciphertexts into ones that the legitimate user can decrypt.

To guarantee the accurate query result, the location information should be stored with the LBS data. Meanwhile, to protect the location privacy of users, the location information should be encrypted as well. In the FINE framework, the location information is encrypted by the variant of CP-AABE described in Section III-A2. The cloud server can decide whether a LBS data file satisfies a LBS query by using the secret key. In order to reduce the computation cost of generation of query on the user side, the user only needs to deal with the second part of $\mathtt{sk}_{\mathbb{S}}$ of the variant of underlying CP-AABE scheme. This method guarantees a constant computation and communication cost on the generation of query no matter how many attributes exist in this system. Usually, the query is related to a location area not a location point. In this case, the user just additionally sends the range to the cloud server, no more computation cost is required. At last, CP-AABE and its variant work together to achieve the confidentiality of access policy in the proposed FINE framework.

For easier reading, we give the description of notation to be used in the FINE framework in Fig. 2.

C. Description of the Proposed FINE Framework

*1) System Initialization (*SI*):* In this phase, the LBS provider chooses a security parameter λ and obtains key pairs (pk, sk) and $(\mathtt{pk}, \mathtt{sk})$ by running $\text{SETUP}(1^\lambda)$ and $\mathtt{Setup}(1^\lambda)$, respectively. In the proposed framework, we consider the dummy attribute corresponding to $(\mathtt{pk}, \mathtt{sk})$ as the location attribute. The LBS provider should keep $\mathtt{T}$ secret. Note that for all key pairs, only $sk, \mathtt{sk}, T, \mathtt{T}$ are different, while $(g_1, g_2, H, e, \mathbb{G}, \mathbb{G}_T)$ are the same.

*2) Service Data Creation (*SDC*):* Before uploading LBS data files to the cloud server, the LBS provider processes them as follows. Assume that the current version number of (pk, sk) and $(\mathtt{pk}, \mathtt{sk})$ is $\mathtt{V}$.

- Select a unique ID for this file;
- Randomly select a symmetric encryption key ek from the key space, for the underlying symmetric key encryption, i.e., AES, and use ek to encrypt the file. The resultant ciphertext is denoted as $C_{ek,file}$.
- Define the access policy $\mathbb{A}$ for the LBS data file.
- Encrypt the symmetric encryption key ek by running $(C_0, \{(C_{i,t_i,0}, C_{i,t_i,1}, C'_{i,t_i,0}, C'_{i,t_i,1})\}) \leftarrow \text{ENC}(\mathbb{A}, pk, ek)$.
- "Encrypt" the location information (L_x, L_y) related to the LBS data file by running $\mathtt{Enc}(\mathbb{A}, \mathtt{pk}, (L_x, L_y))$. The resultant ciphertexts are $(\mathtt{C}_0, \mathtt{C}'_0, \mathtt{C}'_{n+1}, \mathtt{C}'_{n+1}, \{(\mathtt{C}_{i,t_i,0}, \mathtt{C}_{i,t_i,1}, \mathtt{C}'_{i,t_i,0}, \mathtt{C}'_{i,t_i,1})\})$.

Finally, each file is stored on the cloud server in the format as shown in Fig.3, where $\mathtt{V}_f$ equals to $\mathtt{V}$ that shows the version number of the used (pk, sk) and $(\mathtt{pk}, \mathtt{sk})$, and $T_{\mathtt{V}_f}$ is the value contained in pk.

Note that if the LBS data file needs to be modified later, the LBS provider just creates new encrypted LBS data file and sends it to the cloud server with its old file identity ID via an authenticated and secure channel. After receiving the new LBS data files, the cloud server will delete the old one at first, and then store the new one if it exists.

Notation	Description
(L_x, L_y)	location coordinates, with meter as unit
$[\delta_x, \delta'_x]$	range of x-axis
$[\delta_y, \delta'_y]$	range of y-axis
Δ_x	$\delta'_x - \delta_x$
Δ_y	$\delta'_y - \delta_y$
$\mathbb{A}$	LBS data file's access policy
$\mathbb{S}$	user's attribute set
q	a big prime
$\mathbb{G}, \mathbb{G}_T$	bilinear groups with order q
g_1, g_2, e	parameters of bilinear groups
ek	secret key for a symmetric key encryption
(pk, sk)	key pair of CP-AABE, held by LBS provider
$(\mathtt{pk}, \mathtt{sk})$	key pair of the variant of CP-AABE, held by the LBS provider
$sk_{\mathbb{S}}$	secret key related to (pk, sk) and $\mathbb{S}$
$tk_{\mathbb{S}}$	transformation key related to (pk, sk) and $\mathbb{S}$
b	the short ElGamal type secret key
$\mathtt{sk}_{\mathbb{S}}$	secret key related to $(\mathtt{pk}, \mathtt{sk})$ and $\mathbb{S}$
$\mathtt{sk}_{\mathbb{S},1}$	key components unrelated to location in $\mathtt{sk}_{\mathbb{S}}$
$\mathtt{sk}_{\mathbb{S},2}$	key components related to location in $\mathtt{sk}_{\mathbb{S}}$
$\mathtt{sk}_{\mathbb{S},2}{}'$	query key generated by $\mathtt{sk}_{\mathbb{S},2}$ and (L_x, L_y)
$\mathtt{V}$	version number of key pairs
H, F	secure cryptographic hash functions
$C_{ek,file}$	encrypted LBS data file encrypted by ek
UL	user list containing the current users
RKL	revocation key list containing all revocation keys from 1st version to the current version
ID, ID_u	file identity and user's pseudo-identity
$\|x\|$	bit-length of x
$\|\mathbb{X}\|$	bit-length of an element in group $\mathbb{X}$
l	number of received LBS data files satisfying the user's LBS query
l'	number of revoked users
n	number of attributes in the system
N	number of attribute values in the system
$\mathcal{N}$	number of LBS data files stored in the cloud server
T_p	timing of a pairing computation
$T_m^{\mathbb{X}}$	timing of a multiplication in group $\mathbb{X}$
$T_e^{\mathbb{X}}$	timing of an exponentiation in group $\mathbb{X}$

Fig. 2. Notations used in the FINE Framework

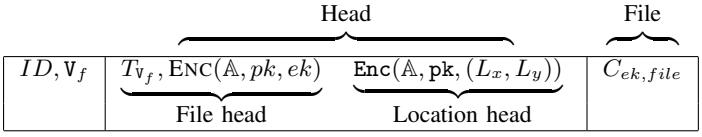

Fig. 3. Format of a file stored on the cloud server

*3) User Grant (*UG*):* When a new user with attribute set $\mathbb{S}$ wants to obtain the location-based services, the LBS provider sends the necessary information to this user and the cloud server as follows. Assume the current version number of (pk, sk) and $(\mathtt{pk}, \mathtt{sk})$ is $\mathtt{V}$.

- Generate secret keys for the user according to her/his attribute set $\mathbb{S}$, i.e., $sk_{\mathbb{S}} \leftarrow \textsc{KeyGen}(sk, \mathbb{S})$ and $\mathtt{sk}_{\mathbb{S}} = (\mathtt{sk}_{\mathbb{S},1}, \mathtt{sk}_{\mathbb{S},2}) \leftarrow \mathtt{KeyGen}(\mathtt{sk}, \mathbb{S})$.
- Generate a transformation key $tk_{\mathbb{S}}$ for the user by raising all components of $sk_{\mathbb{S}}$ to b, where b is a random value from $\mathbb{Z}_q^*$, i.e., $tk_{\mathbb{S}} = sk_{\mathbb{S}}^b = \{(d_{i0}^b, d_{i1}^b, {d'}_{i0}^b, {d'}_{i1}^b)\}$.
- Send $(ID_u, tk_{\mathbb{S}}, \mathtt{sk}_{\mathbb{S},1})$ to the cloud server via an authenticated and secure channel, where $ID_u = F(\mathbb{S}, b)$, F is a cryptographic hash function $\{0,1\}^* \rightarrow \{0,1\}^\ell$, and ℓ is a security parameter. The LBS provider should also store ID_u, which will be used in the user revocation operation.
- Send $(\mathtt{V}_u, b, \mathtt{sk}_{\mathbb{S},2})$ to the user via an authenticated and secure channel, where $\mathtt{V}_u = \mathtt{V}$.

Upon receiving $(ID_u, tk_{\mathbb{S}}, \mathtt{sk}_{\mathbb{S},1})$, the cloud server just adds them into the system user list UL.

After receiving the data from the LBS provider, the user accepts $(\mathtt{V}_u, (b, \mathtt{sk}_{\mathbb{S},2}))$ as the current version number of (pk, sk) and $(\mathtt{pk}, \mathtt{sk})$, and secret key, respectively.

Fig. 4 gives a high-level description of the above process.

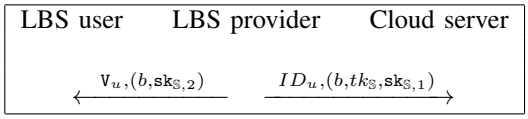

Fig. 4. User grant phase in the proposed FINE framework

*4) User Revocation (*UR*):* Assume that the current version number of (pk, sk) and $(\mathtt{pk}, \mathtt{sk})$ is $j-1$, and the identity of the user to be revoked is ID_u. Note that ID_u could be several identities, if there exist many users to be revoked at the same time. The user revocation is processed as follows.

- The LBS provider chooses random values $\alpha_j, \bar{\alpha}_j$ from $\mathbb{Z}_q^*$ as new master secret keys sk and $\mathtt{sk}$, respectively. It also computes the new public keys as $T = e(g_1, g_2)^{\alpha_j}$ and $\mathtt{T} = e(g_1, g_2)^{\bar{\alpha}_j}$, and the revocation keys as $rk_{j-1,j} = \alpha_{j-1}/\alpha_j$ and $\mathtt{rk}_{j-1,j} = \bar{\alpha}_{j-1}/\bar{\alpha}_j$, where α_{j-1} and $\bar{\alpha}_{j-1}$ are the values of the $(j-1)$-th version sk and $\mathtt{sk}$, respectively.
- The LBS provider sends $(ID_u, j, rk_{j-1,j}, \mathtt{rk}_{j-1,j})$ to the cloud server via an authenticated and secure channel.
- On receiving $(ID_u, j, rk_{j-1,j}, \mathtt{rk}_{j-1,j})$, the cloud server firstly removes the item corresponding to ID_u in the user list UL, and then adds $(j, rk_{j-1,j}, \mathtt{rk}_{j-1,j})$ into the revocation key list RKL.

A high-level description of the user revocation phase is given in Fig. 5.

*5) Location-based Service (*LBS*):* In this phase, the user interacts with the cloud server as follows.

- The user firstly obtains her/his location from her/his mobile device, denoting (L'_x, L'_y) as the location informatoin. Then, s/he computes query key $\mathtt{sk}_{\mathbb{S},2}{}' =$

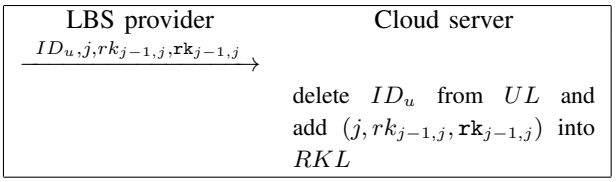

Fig. 5. User revocation phase in the proposed FINE framework

$(\mathtt{d}_{n+1}H(1||g_1)^{L_x}, \mathtt{d}'_{n+1}H(2||g_2)^{L_y})$, where $\mathtt{sk}_{\mathbb{S},2} = (\mathtt{d}_{n+1}, \mathtt{d}'_{n+1})$ is known by the user.

- The user decides the range of location $[\delta_x, \delta'_x]$ and $[\delta_y, \delta'_y]$ s/he wants to query, where $\delta_x, \delta'_x, \delta_y, \delta'_y \in \mathbb{Z}_q$, and s/he wants to obtain the LBS data on the location point (L_x, L_y) satisfying $\delta_x \leq (L_x - L'_x) \bmod q \leq \delta'_x$ and $\delta_y \leq (L_y - L'_y) \bmod q \leq \delta'_y$.
- The user sends $(F(\mathbb{S}, b), \mathtt{V}_u, \delta_x, \delta'_x, \delta_y, \delta'_y, \mathtt{sk}_{\mathbb{S},2}{}')$ to the cloud server via an authenticated and secure channel.
- On receiving the data from the user, the cloud server first checks whether the item related to $F(\mathbb{S}, b)$ exists in the user list UL. If not, the cloud server refuses to provide services to the user; otherwise, it performs the next steps.
- The cloud server performs the following *location test* steps on every file for each value $D_x \in [\delta_x, \delta'_x]$ and $D_y \in [\delta_y, \delta'_y]$.

  1) Use the query key $\mathtt{sk}_{\mathbb{S},2}{}'$ to compute

$$\begin{aligned}\mathtt{C}' = &\left(\prod_{i=1}^{n} \frac{e(\mathtt{d}_{i0}, \mathtt{C}_{i,k_i,1}) e(\mathtt{d}'_{i0}, \mathtt{C}'_{i,k_i,1})}{e(\mathtt{d}_{i1}, \mathtt{C}_{i,k_i,0}) e(\mathtt{d}'_{i1}, \mathtt{C}'_{i,k_i,0})}\right)^{\frac{\bar{\alpha}_{\mathtt{V}_f}}{\bar{\alpha}_{\mathtt{V}_u}}} \cdot \\ & e(\mathtt{d}_{n+1}H(1||g_1)^{L'_x} \cdot H(1||g_1)^{D_x}, \mathtt{C}_{n+1})^{\frac{\bar{\alpha}_{\mathtt{V}_f}}{\bar{\alpha}_{\mathtt{V}_u}}} \cdot \\ & e(\mathtt{d}'_{n+1}H(2||g_2)^{L'_y} \cdot H(2||g_2)^{D_y}, \mathtt{C}'_{n+1})^{\frac{\bar{\alpha}_{\mathtt{V}_f}}{\bar{\alpha}_{\mathtt{V}_u}}}\end{aligned} \tag{1}$$

  The above equation can be re-written as the following according to the correctness of the variant of CP-AABE.

$$\begin{aligned}\mathtt{C}' = \; & (e(g_1, g_2)^{\mathtt{z} \cdot \bar{\alpha}_{\mathtt{V}_u}})^{\frac{\bar{\alpha}_{\mathtt{V}_f}}{\bar{\alpha}_{\mathtt{V}_u}}} \cdot \\ & (e(g_1, H(1||g_1))^{\mathtt{z}_{n+1} \cdot (L'_x + D_x)})^{\frac{\bar{\alpha}_{\mathtt{V}_f}}{\bar{\alpha}_{\mathtt{V}_u}}} \cdot \\ & (e(g_2, H(2||g_2))^{(\mathtt{z} - \mathtt{z}_{n+1}) \cdot (L'_y + D_y)})^{\frac{\bar{\alpha}_{\mathtt{V}_f}}{\bar{\alpha}_{\mathtt{V}_u}}} \\ = \; & e(g_1, g_2)^{\mathtt{z} \cdot \bar{\alpha}_{\mathtt{V}_f}} \cdot \\ & (e(g_1, H(1||g_1))^{\mathtt{z}_{n+1} \cdot (L'_x + D_x)})^{\frac{\bar{\alpha}_{\mathtt{V}_f}}{\bar{\alpha}_{\mathtt{V}_u}}} \cdot \\ & (e(g_2, H(2||g_2))^{(\mathtt{z} - \mathtt{z}_{n+1}) \cdot (L'_y + D_y)})^{\frac{\bar{\alpha}_{\mathtt{V}_f}}{\bar{\alpha}_{\mathtt{V}_u}}}\end{aligned} \tag{2}$$

  where $\bar{\alpha}_{\mathtt{V}_u}$ and $\bar{\alpha}_{\mathtt{V}_f}$ are the values of $\bar{\alpha}$ with version number $\mathtt{V}_u$ and $\mathtt{V}_f$, respectively. Note that assume that the current version number is j, the value of $\frac{\bar{\alpha}_{\mathtt{V}_f}}{\bar{\alpha}_{\mathtt{V}_u}}$ can be computed as $\frac{\bar{\alpha}_{\mathtt{V}_f}}{\bar{\alpha}_{\mathtt{V}_u}} = \frac{\prod_{k=\mathtt{V}_f}^{j-1} \mathtt{rk}_{k,k+1}}{\prod_{k=\mathtt{V}_u}^{j-1} \mathtt{rk}_{k,k+1}}$.

  2) Check whether $\mathtt{C}'/\mathtt{C}_0 = \mathtt{C}'_0{}^{\frac{\bar{\alpha}_{\mathtt{V}_f}}{\bar{\alpha}_{\mathtt{V}_u}}}$ holds. If it holds, then it means that the related LBS data file satisfies the user's query, and go to the next step; otherwise, it chooses another (D_x, D_y) pair and goes back to Step 1). If all $D_x \in [\delta_x, \delta'_x]$ and $D_y \in [\delta_y, \delta'_y]$ have been tested, then go back to Step 1) for another LBS data file. Note that if $L_x = L'_x + D_x$ and $L_y = L'_y + D_x$, we have that

$$\begin{aligned}\mathtt{C}'/\mathtt{C}_0 = & e(g_1, g_2)^{\mathtt{z} \cdot \bar{\alpha}_{\mathtt{V}_f}} / (e(g_1, g_2)^{\bar{\alpha}_{\mathtt{V}_f}})^{\mathtt{z}} \cdot \\ & (e(g_1, H(1||g_1))^{\mathtt{z}_{n+1} \cdot (L'_x + D'_x)})^{\frac{\bar{\alpha}_{\mathtt{V}_f}}{\bar{\alpha}_{\mathtt{V}_u}}} \cdot \\ & (e(g_2, H(2||g_2))^{(\mathtt{z} - \mathtt{z}_{n+1}) \cdot (L'_y + D'_y)})^{\frac{\bar{\alpha}_{\mathtt{V}_f}}{\bar{\alpha}_{\mathtt{V}_u}}} \\ = & (e(g_1, H(1||g_1))^{\mathtt{z}_{n+1} \cdot L_x})^{\frac{\bar{\alpha}_{\mathtt{V}_f}}{\bar{\alpha}_{\mathtt{V}_u}}} \cdot \\ & (e(g_2, H(2||g_2))^{(\mathtt{z} - \mathtt{z}_{n+1}) \cdot L_y})^{\frac{\bar{\alpha}_{\mathtt{V}_f}}{\bar{\alpha}_{\mathtt{V}_u}}} = \mathtt{C}'_0{}^{\frac{\bar{\alpha}_{\mathtt{V}_f}}{\bar{\alpha}_{\mathtt{V}_u}}}\end{aligned} \tag{3}$$

  3) Use the transformation key $tk_{\mathbb{S}}$ to compute

$$C' = \left(\prod_{i=1}^{n} \frac{e(d^b_{i0}, C_{i,k_i,1}) e(d'^b_{i0}, C'_{i,k_i,1})}{e(d^b_{i1}, C_{i,k_i,0}) e(d'^b_{i1}, C'_{i,k_i,0})}\right)^{\frac{\alpha_{\mathtt{V}_f}}{\alpha_{\mathtt{V}_u}}} \tag{4}$$

  The above equation can be written as the following.

$$C' = \left(e(g_1, g_2)^{z \cdot b \cdot \alpha_{\mathtt{V}_u}}\right)^{\frac{\alpha_{\mathtt{V}_f}}{\alpha_{\mathtt{V}_u}}} = e(g_1, g_2)^{z \cdot b \cdot \alpha_{\mathtt{V}_f}} \tag{5}$$

  The above reduction can be obtained according to the correctness of the underlying CP-AABE, and $\alpha_{\mathtt{V}_u}$ and $\alpha_{\mathtt{V}_f}$ are the values of α with version number $\mathtt{V}_u$ and $\mathtt{V}_f$, respectively. Note that assume that the current version number is j, the value of $\frac{\alpha_{\mathtt{V}_f}}{\alpha_{\mathtt{V}_u}}$ can be computed as $\frac{\alpha_{\mathtt{V}_f}}{\alpha_{\mathtt{V}_u}} = \frac{\prod_{k=\mathtt{V}_f}^{j-1} rk_{k,k+1}}{\prod_{k=\mathtt{V}_u}^{j-1} rk_{k,k+1}}$.

- The cloud server sends all satisfied (C')'s along with the corresponding $(T_{\mathtt{V}_f}, C_0, C_{ek,file})$'s to the user via an authenticated and secure channel. Note that (C', C_0) can be considered as an ElGamal type ciphertext with $C' = (e(g_1, g_2)^b)^{z \cdot \alpha_{\mathtt{V}_f}}$ and $C_0 = e(g_1, g_2)^{z \cdot \alpha_{\mathtt{V}_f}}$.
- On receiving all $(T_{\mathtt{V}_f}, C', C_0, C_{ek,file})$'s from the cloud server, the user first gets ek by $C_0/C'^{1/b} = T_{\mathtt{V}_f}{}^z \cdot ek/e(g_1, g_2)^{z \cdot \alpha_{\mathtt{V}_f}} = e(g_1, g_2)^{z \cdot \alpha_{\mathtt{V}_f}} \cdot ek/e(g_1, g_2)^{z \cdot \alpha_{\mathtt{V}_f}} = ek$, and then s/he uses ek to decrypt $C_{ek,file}$.

A high-level description of the location-based service phase is given in Fig. 6.

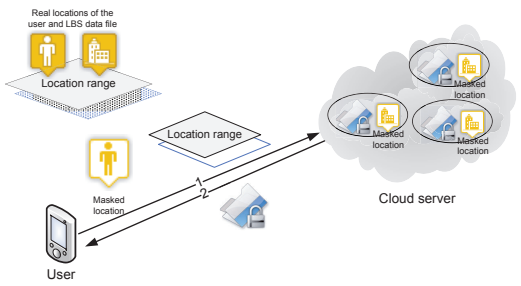

Fig. 6. Location-based service phase in the proposed FINE framework

IV. Security Analysis

In this section, we analyze the required security properties of the proposed FINE framework one by one.

A. Fine-Grained Access Control

In the FINE framework, the LBS provider can define and enforce access policy for each LBS data file and its location information. Only the user whose attribute set satisfies the access policy can obtain the LBS data. Due to the property of CP-AABE, the access policy could be any expressive and flexible access structure.

B. Access Policy Confidentiality

We analyze access policy confidentiality of the FINE framework by comparing it with the standard CP-AABE, which is provably secure under selective ciphertext-policy and chosen message model given the Decision Bilinear Diffie-Hellman assumption and D-Linear assumption [18]. As described in Section III-C, there are two independent kinds of ciphertexts: one is for location head, the other is for file head. Hence, we have the analysis in two parts.

1) Location head: The CP-AABE ciphertext components $\{(\mathtt{C}_{i,t_i,0}, \mathtt{C}_{i,t_i,1}, \mathtt{C}'_{i,t_i,0}, \mathtt{C}'_{i,t_i,1})\}$ and the partial secret key $\mathtt{sk}_{\mathbb{S},1}$ related to location head are computed as that in the standard CP-AABE scheme. On the other hand, $\mathtt{sk}_{\mathbb{S},2}{}'$ is also computed by using the same method of $\mathtt{sk}_{\mathbb{S},1}$. As stated in [18], the ciphertext and secret key related to location head won't reveal the access policy of LBS data. In the FINE framework, cloud server can additionally obtain revocation keys related to location head. However, the values of $\mathtt{T}$'s are kept secret by the LBS provider. Hence, from the viewpoint of the cloud server, the revocation keys are random values, and won't reveal any information of the access policy.

2) File head: As described in Section III-C, the ciphertexts related to file head are computed as that in the standard CP-AABE scheme. As stated in [18], they won't reveal access policy. On the other hand, cloud server can also obtain revocation keys and transformation keys related to file head in the FINE framework. However, the revocation keys only allow the cloud server to translate ciphertexts into other ciphertexts that still won't reveal access policy. The transformation keys enable the cloud server to translate CP-AABE ciphertexts into ElGamal type ciphertexts. Due to the security of ElGamal type ciphertexts, the cloud server without knowing the short ElGamal type secret key b have no idea about whether the resultant ElGamal type ciphertexts is regular or not. Similar analysis is also given in [11]. Hence, the transformation keys do not help the cloud server reveal the access policy.

At last, due to the collusion resistance property of CP-AABE, the cloud server colluding some malicious users cannot obtain any more information of the access policy beyond the fact their secret keys can decrypt the ciphertext or not.

C. LBS data confidentiality

In the proposed FINE framework, the LBS data file is encrypted by a secure symmetric key encryption with a secret key ek, which is further encrypted by CP-AABE. Compared to the standard CP-AABE, the cloud server in the proposed FINE framework can additionally obtain transformation keys, revocation keys. However, the use of transformation key and revocation key cannot hurt the message confidentiality of the underlying public key encryption schemes as analyzed in [11] and [18], respectively.

Readers may notice that the success of user revocation is up to whether the cloud server follows instructions of the FINE framework. In particular, the user revocation won't work if the cloud server continues serving the revoked user, since the cloud server can translate ciphertexts under any version public key into other ciphertexts under any other version public key. While as we stated in Section II-B, the cloud server is assumed to be honest-but-curious, i.e., they should follow the proposed protocol in general. This assumption is accepted in many previously reported papers [16], [20], [21].

D. Location privacy

As described in Section III-C, the location information (L_x, L_y) is only used to compute $\mathtt{C}'_0$ and $\mathtt{sk}_{\mathbb{S},2}{}'$, which may reveal (L_x, L_y). Recall the equation $\mathtt{C}'_0 = e(g_1, H(1||g_1))^{\mathtt{z}_{n+1} \cdot L_x} \cdot e(g_2, H(2||g_2))^{(\mathtt{z}-\mathtt{z}_{n+1}) \cdot L_y} = e(\mathtt{C}_{n+1}, H(1||g_1)^{L_x} \cdot e(\mathtt{C}'_{n+1}, H(2||g_2))^{L_y}$, the cloud server can test every pair (L_x, L_y) whether the equation holds or not. However, when L_x is fixed, there always exists an L_y satisfying the equation if we assume that both L_x space and L_y space are $\mathbb{Z}_q^*$. Hence, for any satisfied pair (L_x, L_y), the cloud server still have no idea whether it is the user's location or not. On the other hand, without knowing $(\mathtt{d}_{n+1}, \mathtt{d}'_{n+1})$, the cloud server cannot obtain (L_x, L_y) directly from $(\mathtt{d}_{n+1} H(1||g_1)^{L_x}, \mathtt{d}'_{n+1} H(2||g_2)^{L_y})$.

There remains another way for the cloud server to get the location information, i.e., keyword guessing attack [22], where the cloud server generates the ciphertexts for all locations, and exhaustively tests which ciphertext with which location information satisfies $\mathtt{sk}_{\mathbb{S},1}$ and $\mathtt{sk}_{\mathbb{S},2}{}'$. However, this attack is stopped at the very beginning. The cloud server cannot generate the ciphertexts without knowing values of $\mathtt{T}$'s, which are kept secret by the LBS provider.

V. Performance Analysis

This section numerically evaluates the performance of the FINE framework in terms of the computation cost on every involved parties as well as the communication cost. The main computation operations in the FINE framework include multiplication and exponentiation in $\mathbb{G}$ and $\mathbb{G}_T$, and pairing.

Note that we do not count the cost on the use of secure and authenticated channel, since this cost for any secure privacy-preserving LBS system is always necessary. For the meaning of notations used in this section, please refer to Fig. 2 in Section III-A.

A. User side

Our main goal on performance is to reduce the computation and communication cost on the user side as much as possible. For easy reading, we summarize the detailed computation cost and communication in Table II, and a high-level summary of cost on the user side is also given in Table III.

TABLE II
COST ON THE USER SIDE IN THE FINE FRAMEWORK

Phase		Computation Cost	Bandwidth Cost
UG		0	$\|\mathbb{V}\| + \|q\| + 2 * \|\mathbb{G}\|$
LBS	Generate	$2T_e^{\mathbb{G}} + 2T_m^{\mathbb{G}}$	$\|ID_u\| + \|\mathbb{V}\| + 4 * \|\mathbb{Z}_q\| + 2 * \|\mathbb{G}\|$
	Receive	$l * (T_e^{\mathbb{G}T} + T_m^{\mathbb{G}T} + T_s)$	$l * (3 * \|\mathbb{G}\| + \|C_{ek,file}\|)$

TABLE III
COMPLEXITY ON THE USER SIDE IN THE FINE FRAMEWORK

Phase	Computation Cost	Bandwidth Cost
UG	0	$\mathcal{O}(1)$
LBS	$\mathcal{O}(l)$	$\mathcal{O}(l)$

B. LBS provider side

The LBS provider is involved in almost all phases except the location-based service phase. The detailed summary is given in Table IV, and a high-level summary of cost on the LBS provider side is given in Table V.

TABLE IV
COST ON THE LBS PROVIDER SIDE IN THE FINE FRAMEWORK

Phase	Computation Cost	Bandwidth Cost
SI	$T_e^{\mathbb{G}}$	0
SDC	$(8*N+2)T_e^{\mathbb{G}}+3T_m^{\mathbb{G}T}+4T_e^{\mathbb{G}T}+2T_p$	$\|ID\|+\|\mathbb{V}\|+4\|\mathbb{G}_T\|+(8*N+2)*\|\mathbb{G}\|$
UG	$(16 * N + 2)T_e^{\mathbb{G}} + 4N * T_m^{\mathbb{G}}$	$\|ID_u\|+\|\mathbb{V}\|+(8*N+2)*\|\mathbb{G}\|+\|q\|$
UR	$2T_e^{\mathbb{G}T} + 2T_m^{\mathbb{Z}q}$	$l' * \|ID_u\| + \|\mathbb{V}\| + 2 * \|q\|$

TABLE V
COMPLEXITY ON THE LBS PROVIDER SIDE IN THE FINE FRAMEWORK

Phase	Computation Cost	Bandwidth Cost
SI	$\mathcal{O}(1)$	0
SDC	$\mathcal{O}(N)$	$\mathcal{O}(N)$
UG	$\mathcal{O}(N)$	$\mathcal{O}(N)$
UR	$\mathcal{O}(1)$	$\mathcal{O}(l')$

C. The cloud server side

The cloud server is involved in almost all phases except the system initialization phase. The detailed summary is given in Table VI, and a high-level summary of cost on the LBS provider side is given in Table VII.

It is worth mentioning that the values of $\left(\prod_{i=1}^{n} \frac{e(\mathtt{d}_{i0},\mathtt{C}_{i,k_i,1})e(\mathtt{d}'_{i0},\mathtt{C}'_{i,k_i,1})}{e(\mathtt{d}_{i1},\mathtt{C}_{i,k_i,0})e(\mathtt{d}'_{i1},\mathtt{C}'_{i,k_i,0})}\right)^{\frac{\tilde{\alpha}_{\mathbb{V}f}}{\tilde{\alpha}_{\mathbb{V}u}}}$ and C' are needed to compute only once for the same user. Hence, the computation cost could be reduced to $2*\mathcal{N}*\Delta_x*\Delta_y$ parings, $3*\mathcal{N}*\Delta_x*\Delta_y$ multiplications in $\mathbb{G}_T$, $4*\mathcal{N}*\Delta_x*\Delta_y$ exponentiations in $\mathbb{G}_T$, $2*\mathcal{N}*\Delta_x*\Delta_y$ multiplications in $\mathbb{G}$ and $2*l$ exponentiations in $\mathbb{G}$. We give a high-level summary of cost on cloud server side in Table VII.

Now we can easy to see that the computation and bandwidth cost on the user side is minimum, which makes the proposed FINE framework is quite suitable for mobile devices.

TABLE VI
COST ON THE CLOUD SERVER SIDE IN THE FINE FRAMEWORK

Phase	Computation Cost	Bandwidth Cost
SDC	0	$\|ID\|+\|\mathbb{V}\|+4*\|\mathbb{G}_T\|+(8*N+2)*\|\mathbb{G}\|$
UG	0	$\|ID_u\| + 8 * N * \|\mathbb{G}\|$
UR	0	$l' * \|ID_u\| + \|\mathbb{V}\| + 2 * \|\mathbb{Z}_q\|$
LBS	$(\mathcal{N}*(4*n+2)*\Delta_x*\Delta_y+4*l*n)*T_p+$ $(\mathcal{N}*(2*n+3)*\Delta_x*\Delta_y+2*l*n)*T_m^{\mathbb{G}T}+$ $(4*\mathcal{N}*\Delta_x*\Delta_y+l)*T_e^{\mathbb{G}T}+$ $(2 * \mathcal{N} * \Delta_x * \Delta_y) * T_m^{\mathbb{G}}+$ $2 * l * T_e^{\mathbb{G}}$	$\|ID\|+\|\mathbb{V}\|+4*\|\mathbb{Z}_q\|+2*\|\mathbb{G}\|+$ $l * (3*\|\mathbb{G}\|+\|C_{ek,file}\|)$

TABLE VII
COMPLEXITY ON THE CLOUD SERVER SIDE IN THE FINE FRAMEWORK

Phase	Computation Cost	Bandwidth Cost
SDC	0	$\mathcal{O}(N)$
UG	0	$\mathcal{O}(N)$
UR	0	$\mathcal{O}(l')$
LBS	$\mathcal{O}(\mathcal{N})$	$\mathcal{O}(l)$

VI. RELATED WORKS

The proposed FINE framework is not only related to privacy-preserving LBS systems, but also related to "access control of outsourced data". In this section, we will review these related works.

A. Privacy-preserving LBS

To provide privacy-preserving LBS, Khoshgozaran and Shahabi [3] proposed an approach where a trusted third party (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. Casper system [4] also requires a TTP named location anonymizer, whose responsibility is to blur user's exact location point into a specified size of cloaked area containing at least $k-1$ other users. To avoid single point of failure, Hashem and Kulik [23] proposed a scheme where the TTP is placed by a group of trusted users constructing an ad-hoc network. However, generating a trusted ad-hoc network in a real world is not always an easy job [6].

To avoid the use of TTP while keeping the location privacy, Kido et al. [9] applied "dummy" locations. In particular, the user should contain many random other locations in her/his LBS query to hide the real location. However, this incurs wasting the energy of mobile devices for choosing, sending and receiving the data related to the fake locations. Ghinita et al. [5] and Paulet et al. [6] made use of private information retrieve (PIR) as a main tool to obtain location privacy without TTP. PIR allows a user to obtain a record from a database without revealing which record s/he is interested in. However, PIR is too huge to be practical for mobile devices[2], since it requires a powerful computational capability. PCQP [7] transforms points of interest in 2-D space into 1-D space with

[2]The journal version of [6] has improved the experimental result on mobile devices, but it still needs 23.9 seconds for one query generation [24].

the Moore's version of Hilbert curve, and resolves the LBS query in the 1-D space with a new secret circular shift scheme. PCQP provides accurate query result and location privacy without TTP; however, the cost on the user side is not only related to the number of the satisfied LBS data files, but also the number of points of interest. This fact shows that PCQP is not suitable for mobile devices when the number of points of interest becomes huge. Jung and Li [8] proposed a privacy-preserving LBS system without TTP by using ABE. However, their solution does not support access policy confidentiality, and brings heavy computation burden to the user side due to the computation of pairings.

B. Access control of outsourced data

By using ABE, PRE and lazy re-encryption, Yu et al. [16] proposed a secure, scalable, and fine-grained data access control system on the cloud server. Our proposed FINE framework looks similar with their system; however, they are different in the following aspects. Yu et al.'s system does not support searchability and minimum cost on the user side. It assumes that the user always knows the file identities of files he wants to retrieves from the cloud server. This assumption cannot be held in the LBS system. Furthermore, we cannot make use of the transformation key to migrate the resource-consuming operation—pairings from users to the cloud server as we do in the proposed FINE framework. It is because that the users in their system should reveal their access trees to the cloud server in order to make the transformation key work. In [21], Wang et al. proposed a mechanism that allows usable and privacy-assured similarity search over outsourced cloud data. However, it cannot support multi-user setting, where LBS works.

VII. Conclusion

In this paper, we have proposed a fine-grained privacy-preserving LBS framework for mobile devices, named FINE. The proposed FINE framework mainly studies how to achieve TTP-freeness, location privacy, confidentiality of the LBS data and its access policy, and fine-grained access control, while keeping the cost on the user side as little as possible. Extensive analysis shows that our proposed FINE framework is secure and highly efficient for mobile devices in terms of computation and communication cost. In our future work, we will carry on the real experiments to further evaluate the effectiveness of the proposed FINE framework. In addition, we will also take it into consideration how to reduce the cost on the cloud server, and how to lower the trust level on the cloud server from honest-but-curious to malicious.

Acknowledgements

Jun Shao was supported in part by NFSC program (No. 61003308 and No. 61100214), NFSZJ No. LR13F02003, QJD1102009, Program for Zhejiang Leading Team of Science and Technology Innovation, and SRF for ROCS, SEM. Rongxing Lu was supported in part by NTU-SUG and MOE AcRF Tier 1 grants.

References

[1] H. Hu, Q. Chen, and J. Xu, "VERDICT: Privacy-preserving authentication of range queries in location-based services," in *ICDE*, IEEE Computer Society, 2013, pp. 1312–1315.

[2] C. Y. Chow, "Privacy-preserving location-based services," Ph.D. dissertation, The University of Minnesota, 2010.

[3] A. Khoshgozaran and C. Shahabi, "Blind evaluation of nearest neighbor queries using space transformation to preserve location privacy," in *SSTD*, Lecture Notes in Computer Science, vol. 4605. Springer, 2007, pp. 239–257.

[4] C.-Y. Chow, M. F. Mokbel, and W. G. Aref, "Casper*: Query processing for location services without compromising privacy," *ACM Transactions on Database Systems*, vol. 34, no. 4, pp. 1–24, Dec. 2009.

[5] G. Ghinita, P. Kalnis, A. Khoshgozaran, C. Shahabi, and K.-L. Tan, "Private queries in location based services: anonymizers are not necessary," in *SIGMOD Conference*, ACM, 2008, pp. 121–132.

[6] R. Paulet, M. G. Kaosar, X. Yi, and E. Bertino, "Privacy-preserving and content-protecting location based queries," in *ICDE*, IEEE Computer Society, 2012, pp. 44–53.

[7] I.-T. Lien, Y.-H. Lin, J.-R. Shieh, and J.-L. Wu, "A novel privacy preserving location-based service protocol with secret circular shift for k-nn search," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 6, pp. 863–873, 2013.

[8] T. Jung and X.-Y. Li, "Search me if you can: Privacy-preserving location query service," *CoRR*, vol. abs/1208.0107, 2012.

[9] H. Kido, Y. Yanagisawa, and T. Satoh, "Protection of location privacy using dummies for location-based services," in *ICDE Workshops*, 2005, p. 1248.

[10] D. Boneh and B. Waters, "Conjunctive, subset, and range queries on encrypted data," in *TCC*, 2007, pp. 535–554.

[11] M. Green, S. Hohenberger, and B. Waters, "Outsourcing the decryption of abe ciphertexts," in *USENIX Security Symposium*, 2011.

[12] R. Lu, X. Lin, and X. Shen, "SPOC: A Secure and Privacy-Preserving Opportunistic Computing Framework for Mobile-Healthcare Emergency," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 3, pp. 614-624, 2013.

[13] M. Blaze, G. Bleumer, and M. Strauss, "Divertible protocols and atomic proxy cryptography," in *EUROCRYPT*,Lecture Notes in Computer Science, Springer, 1998, pp. 127–144.

[14] J. Shao and Z. Cao, "Cca-secure proxy re-encryption without pairings," in *Public Key Cryptography*, Lecture Notes in Computer Science, vol. 5443. Springer, 2009, pp. 357–376.

[15] X. Liang, Z. Cao, H. Lin, and J. Shao, "Attribute based proxy re-encryption with delegating capabilities," in *ASIACCS*, 2009, pp. 276–286.

[16] S. Yu, C. Wang, K. Ren, and W. Lou, "Achieving secure, scalable, and fine-grained data access control in cloud computing," in *INFOCOM*. IEEE, 2010, pp. 534–542.

[17] D. Boneh and M. Franklin, "Identity-based encryption from the weil pairing," *SIAM Journal of Computing*, vol. 32, no. 3, pp. 586–615, 2003.

[18] J. Li, K. Ren, B. Zhu, and Z. Wan, "Privacy-aware attribute-based encryption with user accountability," in *ISC*, Lecture Notes in Computer Science, vol. 5735, 2009, pp. 347–362.

[19] T. ElGamal, "A public key cryptosystem and a signature scheme based on discrete logarithms." *IEEE Transactions on Information Theory*, vol. 31, no. 4, pp. 469–472, 1985.

[20] S. D. C. di Vimercati, S. Foresti, S. Jajodia, S. Paraboschi, and P. Samarati, "Over-encryption: Management of access control evolution on outsourced data," in *VLDB*, ACM, 2007, pp. 123–134.

[21] C. Wang, K. Ren, S. Yu, and K. M. R. Urs, "Achieving usable and privacy-assured similarity search over outsourced cloud data," in *INFOCOM*, IEEE, 2012, pp. 451–459.

[22] J. W. Byun, H. S. Rhee, H.-A. Park, and D. H. Lee, "Off-line keyword guessing attacks on recent keyword search schemes over encrypted data," in *Secure Data Management*, Lecture Notes in Computer Science, vol. 4165. Springer, 2006, pp. 75–83.

[23] T. Hashem and L. Kulik, "Safeguarding location privacy in wireless ad-hoc networks," in *UbiComp*, Lecture Notes in Computer Science, vol. 4717. Springer, 2007, pp. 372–390.

[24] R. Paulet, M. G. Kaosar, X. Yi, and E. Bertino, "Privacy-preserving and content-protecting location based queries," *IEEE Transactions on Knowledge and Data Engineering*, 2013.