sensors

MDPI

*Article*

# Privacy-Preserving Location-Based Service Scheme for Mobile Sensing Data †

**Qingqing Xie [1,‡,§] and Liangmin Wang [2,*,§]**

[1] School of Computer Science and Technology, Anhui University, Hefei 230601, China; xieqn@ahu.edu.cn
[2] School of Computer Science and Communication Engineering, Jiangsu University, Zhenjiang 212013, China
* Correspondence: wanglm@ujs.edu.cn; Tel.: +86-511-8898-6871

† This paper is an extended version of our paper published in Xie, Q.Q.; Wang, L.M. Efficient Privacy-Preserving Processing Scheme for Location-Based Queries in Mobile Cloud. In Proceedings of the IEEE DSC, Changsha, China, 13–16 June 2016 (Accepted).

‡ Current address: Economic and Technology Development Zone, 111 Jiulong Road, Hefei 230601, China.

§ These authors contributed equally to this work.

Academic Editor: Leonhard M. Reindl
Received: 9 August 2016; Accepted: 22 November 2016; Published: 25 November 2016

**Abstract:** With the wide use of mobile sensing application, more and more location-embedded data are collected and stored in mobile clouds, such as iCloud, Samsung cloud, etc. Using these data, the cloud service provider (CSP) can provide location-based service (LBS) for users. However, the mobile cloud is untrustworthy. The privacy concerns force the sensitive locations to be stored on the mobile cloud in an encrypted form. However, this brings a great challenge to utilize these data to provide efficient LBS. To solve this problem, we propose a privacy-preserving LBS scheme for mobile sensing data, based on the RSA (for Rivest, Shamir and Adleman) algorithm and ciphertext policy attribute-based encryption (CP-ABE) scheme. The mobile cloud can perform location distance computing and comparison efficiently for authorized users, without location privacy leakage. In the end, theoretical security analysis and experimental evaluation demonstrate that our scheme is secure against the chosen plaintext attack (CPA) and efficient enough for practical applications in terms of user side computation overhead.

**Keywords:** mobile sensing; mobile cloud; location-based service; privacy preservation

## 1. Introduction

Recently, mobile sensing devices have been widely used in data sensing [1,2], including location data [3,4]. For example, when a person takes photos by a smart phone, the equipped location sensor (GPS modules) can always acquire the locations where the photos are taken [5]. Additionally, the locations are embedded into the photos for remembrance. Then, these location-embedded data will be published in the mobile cloud automatically [1,2,6], such as iCloud, Samsung cloud, etc. These location-embedded data bring great convenience for cloud service providers (CSP) to provide location-based services (LBS) for users [3,7].

However, the mobile cloud is untrustworthy. Additionally, the location itself contains much personal information [8–10]. CSP is curious to infer and analyze location data to harvest additional information to gain illegal profits. Thus, the publisher (i.e., data owner) requires a solution that can protect location privacy from unauthorized users and CSP. As shown in Figure 1, Alice takes some food photos by iPhone, and the photos are embedded with location information. She stores them in iCloud and shares them with her friends. Since the location where the photos are taken is her home, she hopes that the embedded location is visible only to her friends, while invisible to strangers.

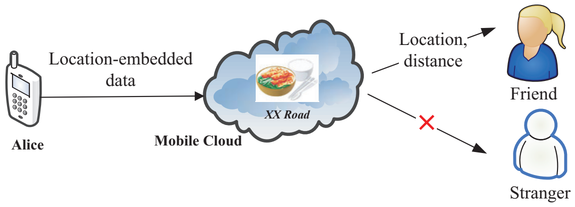


**Figure 1.** An example in which Alice shares location-embedded data with her friends.

A naive solution is to store the location data on the mobile cloud in an encrypted format. To achieve secure sharing on encrypted data, some researchers have studied the attribute-based encryption (ABE) scheme [4,11–14]. In this scheme, the publisher encrypts the confidential location information using a symmetric encryption scheme (AES or DES) and defines the access policies, then uploads the encrypted location and access polices into the cloud for storage. Only authorized queriers (whose attributes satisfy the access policies) can read the location data. During the whole procedure, the cloud undertakes only storage overhead. However, in LBS, the queriers also require the CSP to provide the services of location distance compute and compare. By applying ABE directly, the authorized queriers cannot process the location until downloading and decrypting the encrypted location data. It takes the queriers too much local storage and computation cost, which is unacceptable considering the weak power of smart phones.

To support functional processing over encrypted data, homomorphic encryption (HE) [15–18] was proposed. Xie and Wang [4] applied the RSA (for Rivest, Shamir and Adleman) algorithm to achieve computational functions on encrypted location data. In addition, Paillier's cryptosystem [19–23] is widely used as HE, due to its high efficiency and simplicity. It involves only one multiplication for each homomorphic addition and one exponentiation for each homomorphic multiplication. Li and Jung [5] combined the ciphertext-policy attribute-based encryption (CP-ABE) with Paillier's cryptosystem to exert fine-grained access control over LBS. One cannot gain any information from the query if his/her identity attributes do not satisfy the access policy defined by the data publisher. In addition, the location distance computed over encrypted location is supported. However, the publisher must stay online to interact with queriers once requested. This is not practical, considering the limited power of smart phone.

To overcome the above problems, we propose a privacy-preserving LBS scheme for mobile sensing data. Here, a new encryption method on the basis of the RSA algorithm (The RSA algorithm is a commonly adopted public key cryptography algorithm. It is named after the three mathematicians who developed it: Rivest, Shamir and Adleman. The security of this encryption algorithm is based on the hardness of the factoring problem. We will present this algorithm in Section 3.2.) and CP-ABE scheme is designed. Our proposed scheme has two advantages as follows:

- Secure sharing over location information with certain queriers. Our scheme achieves that location information is visible to specific queriers, while kept secret from others.
- Efficient and privacy-preserving location distance compute and compare. The location distance compute and compare are two of the most common functions in LBQ, i.e., what is the distance between the publisher's and querier's locations, or whether the distance is less than 100 m. Compared with the privacy-preserving location query protocol (PLQP) scheme [5], we make better use of the powerful energy in the mobile cloud, by the mobile cloud undertaking most of the computing overhead, such that the computation cost at the querier is very low.

The main contributions of our paper are outlined as follows:

1. A novel mobile sensing service system is constructed for privacy-preserving LBS.
2. This paper designs a novel encryption method on the basis of the RSA algorithm and CP-ABE scheme, so that the mobile cloud can process LBS over encrypted location information and only authorized queriers can get the query results.

The rest of this paper is organized as follows. Section 2 discusses the related work. Section 3 presents some preliminaries. Section 4 describes our system models. Section 5 gives the detailed design of our privacy-preserving LBS scheme for mobile sensing data. Section 6 analyzes the security of our proposed scheme. Section 7 shows the performance evaluation by experiments. Finally, Section 8 concludes this paper.

## 2. Related Work

Our paper designs a privacy-preserving LBS scheme for mobile sensing data. The related work mainly includes two aspects, i.e., privacy-preserving LBS and the access control technique.

### 2.1. Privacy-Preserving LBS

The k-anonymity technique has been widely used to achieve user location privacy in LBS. The basic idea is to remove some features, such that each item is not distinguishable among other $k - 1$ items. It can ensure that a user can be identified with a probability of at most $1/k$.

Kido et al. [24] proposed an anonymous communication technique for LBS to protect location privacy using dummies. Duckham and Kulik [25] presented a privacy-preserving location query algorithm by using the obfuscation method and vague location information of the user. Chow et al. [26] proposed a distributed k-anonymity model and a peer-to-peer spatial cloaking algorithm for the anonymous location-based services. Mokbel [27] proposed a location-obfuscation method that allows the server to record the real identifier of the user, but decreases the precision of the location information to protect the location privacy. Bamba et al. [28] proposed fast and effective location cloaking algorithms for location k-anonymity and location l-diversity in a mobile environment. Gedik and Liu [29] applied the personalized k-anonymity model to protect the location privacy of the user. Shankar et al. [30] proposed a fully-decentralized and autonomous k-anonymity-based system for location-based queries. Xue et al. [31] introduced the concept of location diversity, which improves spatial k-anonymity by ensuring that each query can be associated with at least $l$ different semantic locations.

All of the above solutions can be applied to LBS. However, their techniques do not allow the cloud to search encrypted data. Therefore, they cannot be used for outsourced LBS where LBS data in the cloud are encrypted.

Li and Jung [5] designed a suite of fine-grained privacy-preserving location query protocols (PLQP) by applying Paillier's cryptosystem [32,33]. It can solve the privacy issues in existing LBS applications. However, once there is an LBS request, the PLQP needs very frequent interaction between the publisher and the querier and much computation cost. In mobile sensing service systems, most queriers access the social networks via smart phones. The smart phones have weak power. Hence, it is unacceptable for the publishers to stay online always.

Shao, Lu and Lin [8] proposed a FINEframework based on the CP-ABE scheme. In this framework, LBS data are outsourced to a cloud server after encryption. Although the framework can ensure the confidentiality of LBS data, their search patterns will lead to the leakage of user location privacy, because their trapdoors generated from the locations are steady, which means trapdoors are always the same for the same location. It is easy for an attacker to count the frequency of a specific trapdoor and identify the known locations. In addition, this method is not efficient due to the low efficiency of the public encryption.

### 2.2. Access Control

Recently, the ABE scheme has been widely used to exert access control for LBS in the mobile cloud. Li and Jung [5] introduced CP-ABE to exert fine-grained access control over the location queries. One cannot gain any information from the query if his/her identity attributes do not satisfy the access policy defined by the data publisher. Shao, Lu and Lin [8] also employed CP-ABE in designing their FINE framework to achieve fine-grained access control over location-based service data.

The ABE scheme enables fine-grained access control over encrypted data using access policies and associates attributes with private keys and ciphertexts [34–37]. It was first proposed by Sahai and Waters [38], later extended to the key-policy ABE (KP-ABE) by Goyal et al. [39] and the CP-ABE by Bethencourt et al. [40]. In KP-ABE, the ciphertext is associated with an attribute set, and the user secret key is associated with an access policy over attributes. The user can decrypt the ciphertext if and only if the attribute set of the ciphertext satisfies the access policy specified in his/her secret key. The encryptor exerts no control over who has access to the data that he/she encrypts. In CP-ABE, the ciphertext is associated with an access policy over attributes, and the user secret key is associated with an attribute set. The user can decrypt the ciphertext if and only if the attribute set of his/her secret key satisfies the access policy specified in the ciphertext. The encryptor is able to decide who should or should not have access to the data that he/she encrypts. In our system model, CP-ABE is more suitable than KP-ABE because it enables the data publishers to determine an access policy over the outsourced location data, as studied by Li and Jung in [5].

## 3. Preliminaries

This section briefly describes some preliminaries used in our work, including the bilinear map, the RSA algorithm, CP-ABE and the access tree.

### 3.1. Bilinear Map

Let $\mathbb{G}_0$ be a multiplicative cyclic group of prime order $p$ and $g_0$ be its generator. The bilinear map $e$ is defined as: $e : \mathbb{G}_0 \times \mathbb{G}_0 \to \mathbb{G}_T$, where $\mathbb{G}_T$ is the codomain of $e$. The bilinear map $e$ has the following properties:

- Bilinearity: $\forall u, v \in \mathbb{G}_0$ and $a, b \in \mathbb{Z}_p$, $e\left(u^a, v^b\right) = e(u, v)^{ab}$
- Symmetry: $\forall u, v \in \mathbb{G}_0$, $e\left(u, v\right) = e\left(v, u\right)$.
- Non-degeneracy: $e\left(g_0, g_0\right) \neq 1$.

**Definition 1** (discrete logarithm assumption)**.** *The discrete logarithm assumption in group $\mathbb{G}_0$ of prime order $p$ with generator $g_0$ is defined as follows: for any probabilistic polynomial-time (PPT) algorithm $\mathcal{A}$, the probability that* $\Pr\left[\mathcal{A}\left(g_0, g_0^a\right) = a\right]$ *is negligible, where $g_0, g_0^a \in \mathbb{G}_0$, and $a \in \mathbb{Z}_p$.*

**Definition 2** (decisional Diffie–Hellman (DDH) problem)**.** *The decisional Diffie–Hellman (DDH) problem in group $\mathbb{G}_0$ of prime order $p$ with generator $g_0$ is defined as follows: on input $g_0, g_0^a, g_0^b, g_0^c = g_0^{ab} \in \mathbb{G}_0$, where $a, b, c \in \mathbb{Z}_p$, decide whether $c = ab$ or $c$ is a random element.*

**Definition 3** (decisional bilinear Diffie–Hellman (DBDH) problem)**.** *The decisional bilinear Diffie–Hellman (DBDH) problem in group $\mathbb{G}_0$ of prime order $p$ with generator $g_0$ is defined as follows: on input $g_0, g_0^a, g_0^b, g_0^c = g_0^{ab} \in \mathbb{G}_0$ and $e(g_0, g_0)^z = e(g_0, g_0)^{abc} \in \mathbb{G}_T$, where $a, b, c \in \mathbb{Z}_p$, decide whether $z = abc$ or $z$ is a random element.*

The security of many ABE schemes relies on the discrete logarithm assumption. The research also assumes that no PPT algorithm can solve the DDH and DBDH problems with non-negligible advantage. This assumption is reasonable since in a large number field, it is widely recognized that discrete logarithm problems (DLP) are as hard as described in Definition 1. Therefore, $a$ is not deducible from $g_0^a$, even if $g_0$ is publicly known.

### 3.2. RSA: Public Key Cryptography Algorithm

RSA is a commonly-adopted public key cryptography algorithm [41]. It is the first and still most widely-used asymmetric algorithm. RSA is named after the three mathematicians who developed it, Rivest, Shamir and Adleman. The public/private key pair of RSA is computed in Algorithm 1, where

$GenModulus\left(1^N\right)$ is a function used to output a composite modulus $n$ along with its two $N$-bit prime factors; $\phi$ is Euler's totient function; $gcd$ is a function used to compute the greatest common divisor for two numbers.

The RSA encryption scheme includes three algorithms as follows:

1. $KeyGen\left(1^N\right) \to pk, sk$: takes security parameter $1^N$ as input and outputs a public/private key pair, denoted as $pk = (n, e)$ and $sk = (p, q, d)$, respectively, by executing Algorithm 1.
2. $Enc\,(m, pk) \to c$: on input, a public key $pk = (n, e)$ and a message $m \in \mathbb{Z}_n^*$ compute the ciphertext as $c = m^e \bmod n$.
3. $Dec\,(c, sk) \to m$: on input, a private key $sk = (p, q, d)$ and a ciphertext $c \in \mathbb{Z}_n^*$ compute the message as $m = c^d \bmod n$.

The security of the RSA encryption scheme relies on the hardness of the factoring problem. If an adversary can factorize $n$, then he/she can compute $\phi\,(n) = (p-1)\,(q-1)$ and obtain the secret key $d$ by utilizing the Euclidean algorithm. However, factoring a large number is still a hard problem. The proper choice of the modulus $n = pq$ can guarantee the security of RSA encryption scheme.

**Algorithm 1** KeyGen.

```
Input:
    $1^N$: security parameter;
Output:
    $pk, sk$: a public/private key pair;
$(n, p, q) \leftarrow GenModulus\left(1^N\right)$;
$\phi\,(n) = (p-1)\,(q-1)$;
Find $e$, such that $gcd\,(e, \phi\,(n)) = 1$, where $1 < e < \phi\,(n)$;
Compute $d = e^{-1} \bmod \phi\,(n)$
    return $pk = (n, e), sk = (p, q, d)$.
```

### 3.3. CP-ABE

In the CP-ABE, the private key is distributed to users by the trusted authority (TA) only once. The keys are identified with a set of descriptive attributes, and the encryptor specifies an encryption policy using an access tree, so that those with private keys the satisfy it can decrypt the ciphertext.

### 3.4. Access Tree $T_P$

In CP-ABE, the encryption policy is described with a tree called access tree $T_p$. Each non-leaf node of the tree is a threshold gate, and each leaf node is described by an attribute. An example is shown in Figure 2.

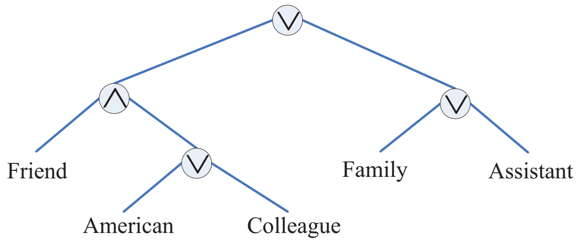


**Figure 2.** An example of the access tree.

In this paper, a publisher's location information is set visible to certain kinds of users. For example, in Figure 1, Alice's location information is only accessible to her friends. If a user's attributes satisfy

$T_P$, he/she is granted with the access privilege. Simultaneously, he/she also can obtain the results of LBS provided by the mobile cloud. By doing so, we can control the visibility of the publisher's location information.

Given a node $u$ in the $T_P$, $|Children(u)|$ is the number of the node $u$'s children nodes, and $k_u$ is its threshold value $0 < k_u \leq |Children(u)|$. The node $u$ is assigned a true value if at least $k_u$ child nodes have been assigned a true value. Specially, the node becomes an $OR$gate when $k_u = 1$ or an $AND$gate when $k_u = |Children(u)|$.

The access tree is described by a set of polynomials, as shown in Algorithm 2. In the access tree $T_P$, the node value of the gate is recovered if and only if the values of at least $k_u$ child nodes are recovered, which is performed in a recursive manner. The notations for the access tree is explained in Table 1.

**Algorithm 2** Access Tree Description.

```
Input:
    $T_P$: an access tree;
Output:
    $\{s, q_{lf}(0) | lf$ is a leaf node of $T_P\}$: $s \in \mathbb{Z}_p$ is a randomly-picked secret integer;
for all $u$ in $T_P$ do
    define a polynomial $q_u(x) = \sum_{i=0}^{k_u-1} a_{ui} x^i$, where the coefficients $a_{ui}$ are undetermined;
end for
Pick a random integer $s$
Set $a_{R0} = q_R(0) = s$, where $R$ is the root node of $T_P$;
Set other coefficients of $q_R(x)$, i.e., $a_{Ri}, i = 1, 2, \ldots, k_R - 1$, by randomly-picked secret integers;
From top to bottom, set all of the coefficients of other nodes (except for the root node) that satisfy the following equation;
    $$q_u(0) = q_{parent_{(u)}}(index(u)).$$
return $\{s, q_{lf}(0) | lf$ is a leaf node of $T_p\}$.
```

**Table 1.** The notations for access tree $T_P$.

| Notation | Description |
|---:|---|
| $u$ | a node in $T_P$ |
| $leaf(T_p)$ | the set of leaf nodes in $T_P$ |
| $Children(u)$ | the set of all the child nodes of $u$ |
| $\|Children(u)\|$ | the number of the node $u$'s child nodes |
| $k_u$ | the threshold value of node $u$ |
| $q_u(x) = \sum_{i=0}^{k_u-1} a_{ui} x^i$ | the polynomial equation of node $u$, where $a_{ui} \in \mathbb{Z}_p, i = 1, 2, ..., k_u - 1$ are the coefficients |
| $parant(u)$ | $u$'s parent node |
| $index(u)$ | the index of node $u$ |

## 4. System Model, Threat Model, Location Assumption and Problem Formulation

### 4.1. System Model

Our mobile sensing service system mainly consists of four entities, as shown in Figure 3: mobile cloud, a publisher, many queriers and TA.

The mobile cloud provides LBS via mobile applications or social network applications based on the collected location data. Its main work is to store and process ciphertext.

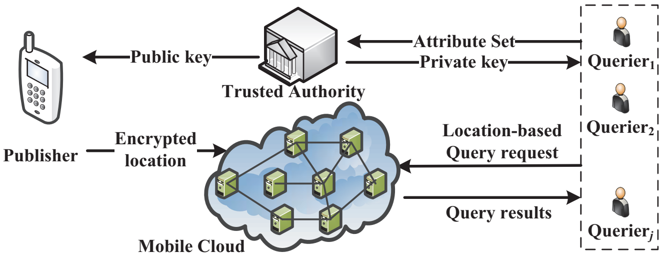


**Figure 3.** The mobile sensing service system.

A publisher contributes his/her location data to the mobile cloud, via smart phone, iPad, etc. Before uploading the data, the publisher first obtains the public key from the TA and determines the access tree. Then, he/she uses the public key and access tree to encrypt his/her location information. Afterwards, the encrypted location information is uploaded to the mobile cloud for storage and sharing. In addition, we assume the origin of a packet is successfully hidden, which is out of this paper's scope, and can be trivially prevented by employing anonymized network protocols [42].

Many queriers submit LBS query requests to the mobile cloud over the collected cloud data. However, only authorized queriers can obtain plain query results.

TA is assumed to have powerful computation abilities. At the setup phase, the TA computes its own master key and the system-wide public parameter. The master key is used to generate the private key for the queriers, and the public key is used to process system-wide operations.

### 4.2. Threat Model

We assume that the mobile cloud is "honest but curious". Specifically, it acts in an "honest" fashion and correctly follows the designated protocol specification. However, it is "curious" to infer and analyze the stored data and queriers' query requests to harvest additional information to gain illegal profits.

The queriers are curious about the confidential information, which is outside of their privileges. They may also collude with the mobile cloud.

### 4.3. Location Assumption

As described in [5], the ground surface can be assumed to be a plane, and every user's location is mapped to an Euclidean space with integer coordinates (with meters as the unit). The Euclidean distance between two locations $X = (x_1, x_2, x_3)$ and $Y = (y_1, y_2, y_3)$ is computed as:

$$dist(X,Y) = |X - Y| = \sqrt{\sum_{i=1}^{3} (x_i - y_i)^2}.$$

As for a real location on the surface of the Earth, we know that the relationship between the surface distance $SD(X,Y)$ and the Euclidean distance $dist(X,Y)$ is as follows:

$$SD\left(X,Y\right) = 2\arcsin\left(\frac{dist\left(X,Y\right)}{2R}\right) \cdot R,$$

where the Earth is assume to be a sphere with radius $R$ meters. Hence, it is easy to compute $SD(X,Y)$ from $dist(X,Y)$. To check if the surface distance satisfies certain conditions, we can convert it to check if the Euclidean distance is satisfying the corresponding conditions. For example, $dist(X,Y) \leq \tau$ is equivalent as:

$$SD(X,Y) \leq 2R\arcsin(\tau/2R).$$

For brevity, in this paper, we will focus on the Euclidean distance instead of the surface distance.

### 4.4. Problem Formulation

Assume that the querier $\mathcal{Q}$'s location information is $X = (x_1, x_2, x_3)$, and the publisher $\mathcal{P}$'s location information $Y = (y_1, y_2, y_3)$ is embedded in the published data. According to $Q$'s attributes set $S_Q$, the publisher $\mathcal{P}$ determines whether the querier $\mathcal{Q}$ can enjoy the LBS related to $\mathcal{P}$'s location. Afterwards, the authorized querier will obtain the corresponding LBS results provided by the mobile cloud.

In this paper, we design a privacy-preserving LBS scheme for mobile sensing data, where the location publisher can determine who can decrypt the ciphered LBS results provided by the mobile cloud. Moreover, no confidential location information is leaked to the mobile cloud and unauthorized users during the LBS processing. Here, the LBS mainly includes two basic types: location distance compute and compare. The proposed scheme includes five main algorithms, as follows:

- $Setup(1^N) \rightarrow MK, PK$
  This algorithm takes a security parameter $1^N$ as input. The TA executes this algorithm to compute its own master key $MK$ and a system-wide public parameter $PK$.
- $Encrypt\,(PK, Y, T_P, K_Y) \rightarrow Y_e$
  This algorithm takes as input the public parameter $PK$, the publisher's location information $Y = (y_1, y_2, y_3)$, an access tree $T_P$ determined by the publisher and an encryption key $K_Y$. It will encrypt the location $Y$, so that a querier can enjoy the LBS over location $Y$ if and only if his/her attributes satisfy the access tree $T_P$.
- $KeyGenerate\,(MK, PK, S_Q) \rightarrow SK_Q$
  This algorithm takes as input the TA's master key $MK$, the public parameter $PK$ and a querier's attribute set $S_Q$. It enables the querier to interact with the TA and to obtain a secret key $SK_Q$.
- $Verify\,(PK, SK_Q, S_Q, Y_e) \rightarrow W^s$ or $\perp$
  This algorithm enables an authorized querier to obtain a critical secret parameter $W^s$, which is the key to decrypt the ciphered query results provided by the mobile cloud.
- $Operate(Y_e, W^s, X) \rightarrow answer$
  In this protocol, firstly, a querier encrypts his/her location $X$ as $X_e$ using $W^s$, then the mobile cloud operates over the encrypted locations $Y_e$ and $X_e$ to compute a ciphered query result. In the end, the querier uses $W^s$ to decrypt the ciphered result as *answer*.

## 5. Our Proposed Scheme

In this section, we will present the scheme design in detail.

A. $Setup(1^N) \rightarrow MK, PK$

This algorithm takes a security parameter $1^N$ as the input, and gives the TA's master key $MK$ and a system-wide public parameter $PK$ as the output, as shown in Algorithm 3.

By Algorithm 3, the TA chooses and publishes a bilinear group $\mathbb{G}_0$ of prime order $p$ with generator $g_0$, then randomly and secretly picks $v_0 \in \mathbb{Z}_p$. Finally, the TA computes the master key $MK$ and the public key $PK$.

B. $Encrypt\,(PK, Y, T_P, K_Y) \rightarrow Y_e$

Before uploading the location data, the publisher executes this algorithm to encrypt the sensitive location information $Y = (y_1, y_2, y_3)$. In addition, she/he determines the access tree $T_P$ to exert access control on the location information. The ciphertext $Y_e$ includes two parts: $Y_e^I$ and $Y_e^{II}$. The encryption procedure consists of three main steps.

1. Pick a symmetric encryption key $K_Y = \langle x, m \rangle$, where $0 < x, m < n$, $n$ is generated by $GenModulus\,(1^N)$, as shown in Section 3.2.
2. Compute:

$$y_{ei} = y_i \cdot \left(mg^{(x \bmod \phi(n))}\right)^{-1} modn, i = 1, 2, 3. \tag{1}$$

Here, $g$ is co-prime with $n$; $\phi(n)$ is Euler's totient function of $n$. We will omit "mod$\phi(n)$" in the following expressions with an assumption that the exponent of the above formula is computed in modular $\phi(n)$.

3. Execute Algorithm 2, and obtain $\{s, q_{lf}(0) | lf \text{ is a leaf node of } T_p\}$. Then, $Y_e^I$ and $Y_e^{II}$ are computed by Equations (2) and (3). In $Y_e^{II}$, $C_u$ and $C'_u$ represent the attribute values in the specified access tree.

$$Y_e^I = \{\langle y_{e1}, y_{e2}, y_{e3}\rangle, \langle x + W^s, m \cdot W^s\rangle\}, \tag{2}$$

$$Y_e^{II} = \langle T_p, \left\{C_u = g_0^{q_u(0)}, C'_u = H(att\,(u))^{q_u(0)}\right\}_{u \in leaf(T_P)}, C = g_0^s\rangle. \tag{3}$$

Finally, the publisher stores $Y_e = \{Y_e^I, Y_e^{II}\}$ in the mobile cloud for sharing them with some queriers. In this system, $Y_e$ can be downloaded by every querier. However, only authorized queriers can obtain the plain location information $Y$, and the query results of location distance compute and compare.

**Algorithm 3** Setup.

```
Input:
    $1^N$: security parameter;
Output:
    $MK$: TA's master key;
    $PK$: public parameter;
1: Choose a bilinear group $\mathbb{G}_0$ of prime order $p$ with generator $g_0$;
2: Pick $v_0 \in \mathbb{Z}_p$ randomly and secretly;
3: Compute the master key $MK = g_0^{v_0}$;
4: Compute $W = e(g_0, g_0)^{v_0}$;
5: Set $PK = \{\mathbb{G}_0, g_0, W\}$;
   return $MK, PK$.
```

C. *KeyGenerate* $(MK, PK, S_Q) \rightarrow SK_Q$

When a new querier $Q$, with attribute set $S_Q$, requests to join the system, TA executes this algorithm to generate $Q$'s secret key. This algorithm is composed of two steps, as follows:

1. Attribute key generate: TA randomly picks $d_0 \in \mathbb{Z}_p$ and computes:

$$D = MK \cdot g_0^{d_0}. \tag{4}$$

For any attribute $i \in S_Q$, TA randomly picks $r_i \in \mathbb{Z}_p$ and computes the partial private key as:

$$D_i = H(i)^{r_i} \cdot g_0^{d_0}, \tag{5}$$

$$D'_i = g_0^{r_i}. \tag{6}$$

where $H(i)$ is the hash value of attribute $i$.

2. Key aggregate: The secret key is generated by aggregating $D$, $D_i$, and $D'_i$ as:

$$SK_Q = \left\{D, \forall i \in S_Q : D_i, D'_i\right\}. \tag{7}$$

The above procedures are described in Algorithm 4.

**Algorithm 4** KeyGenerate.

```
Input:
    $MK$: TA's master key;
    $PK$: the public parameter;
    $S_Q$: a querier's attribute set;
Output:
    $SK_Q$: a secret key;
    Compute $D = MK \cdot g_0^{d_0}$, where $d_0 \in \mathbb{Z}_p$;
    for all $i \in S_Q$ do
        Compute $D_i = H(i)^{r_i} \cdot g_0^{d_0}$, $D'_i = g_0^{r_i}$, where $r_i \in \mathbb{Z}_p$;
    end for
    return $SK_Q = \{D, \forall i \in S_Q : D_i, D'_i\}$.
```

D. $Verify\,(PK, SK_Q, S_Q, Y_e) \rightarrow W^s$ or $\perp$

By executing this algorithm, only authorized querier can obtain the secret parameter $W^s$, which will be used to decrypt the ciphered query results. Otherwise, it will output $\perp$. Figure 4 shows the main overview of this algorithm.

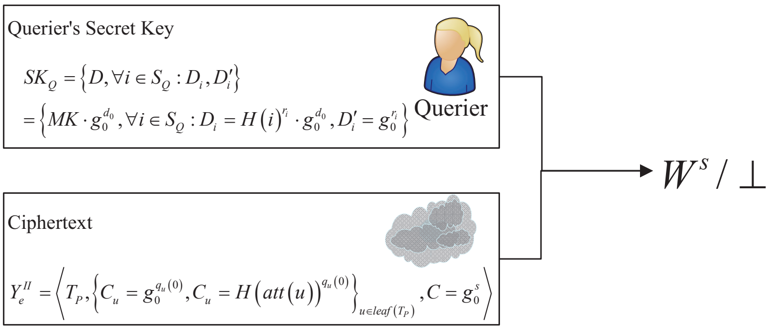

Querier's Secret Key
$SK_Q = \{D, \forall i \in S_Q : D_i, D'_i\}$
$= \{MK \cdot g_0^{d_0}, \forall i \in S_Q : D_i = H(i)^{r_i} \cdot g_0^{d_0}, D'_i = g_0^{r_i}\}$
Querier
Ciphertext
$Y_e^{II} = \left\langle T_P, \left\{C_u = g_0^{q_u(0)}, C_u = H(att(u))^{q_u(0)}\right\}_{u \in leaf(T_P)}, C = g_0^s \right\rangle$
$W^s / \perp$

**Figure 4.** The overview of verification process.

Firstly, a recursive algorithm $DecryptNode\,(Y_e^{II}, SK_Q, S_Q, u)$ is defined as Algorithm 5, where $u$ stands for a node in the access tree $T_P$.

Then, the querier recursively calls $DecryptNode\,(Y_e^{II}, SK_Q, S_Q, R)$ from the root node $R$, and obtains $par_R = e(g_0, g_0)^{s \cdot d_0}$. At last, he/she can get the secret output $W^s$ by computing Equation (8).

$$\frac{e\,(C, D)}{par_R} = W^s \tag{8}$$

E. $Operate(Y_e, W^s, X) \rightarrow answer$

In this protocol, firstly, the authorized querier uses $W^s$ to encrypt his/her location $X$ as $X_e$, then the mobile cloud operates over $Y_e$ and $X_e$ to compute a ciphered location distance or to test whether the distance between these two locations is far or not. If necessary, the querier will use $W^s$ to decrypt the ciphered result as $answer$.

In our scheme, we consider these two types of operations: location distance compute and location distance compare, i.e., what the distance between these two locations is or whether the distance is far or not. They are two basic LBS.

**Algorithm 5** DecryptNode.

```
Input:
    $Y_e^{II}$: defined in Equation (3);
    $SK_Q$: the querier $Q$'s secret key;
    $S_Q$: the querier $Q$'s attribute set;
    $u$: a node in the access tree $T_P$;
Output:
    $par_u$: a secret parameter;
    or $\perp$
if $u$ is a leaf node then
    Set $i = att(u)$;
    if $i \in S_Q$ then
        Compute
            $$par_u = \frac{e(D_i, C_u)}{e(D'_i, C'_u)} = e(g, g)^{q_u(0) \sum d_k};$$
    else return $par_u = \perp$;
    end if
else
    Define $F_u = null$;
    for all $z \in Children(u)$ do
        Compute $par_z = DecryptNode\left(Y_e^{II}, SK_Q, S_Q, z\right)$;
        if $par_z \neq \perp$ then
            Update $F_u = F_u \cup \{par_z\}$;
        end if
    end for
    if $|F_u| < k_u$ then return $par_u = \perp$;
    else
        Compute $par_u = e(g_0, g_0)^{q_u(0) d_0}$ using $F_u$ by polynomial interpolation method;
    end if
end if
return $par_u$.
```

Here, the locations of the publisher and the querier are $X{=}(x_1, x_2, x_3)$ and $Y{=}(y_1, y_2, y_3)$, respectively. Additionally, the publisher's location data $Y{=}(y_1, y_2, y_3)$ have been encrypted as $Y_e{=}\left\{Y_e^I, Y_e^{II}\right\}$ by Equations (2) and (3) and stored in the mobile cloud. The querier encrypts his/her location $X$ using Equation (9).

$$\begin{aligned} X_e &= (x_{e1}, x_{e2}, x_{e3}) \\ &= (x_1, x_2, x_3) \cdot W^s \cdot g^{W^s} \\ &= \left(x_1 \cdot W^s \cdot g^{W^s}, x_2 \cdot W^s \cdot g^{W^s}, x_3 \cdot W^s \cdot g^{W^s}\right) \end{aligned} \tag{9}$$

Next, we will present how to perform these two operations.

1. Location distance compute:

We know that the distance between the publisher and the querier can be computed as:

$$dis = \sqrt{\sum_{i=1}^{3} (x_i - y_i)^2}. \tag{10}$$

The querier encrypts his/her location $X = (x_1, x_2, x_3)$ as $X_e$ by Equation (9) and sends $X_e$ to the mobile cloud. Then, the mobile cloud executes Algorithm 6 to compute the ciphered location distance between $X_e$ and $Y_e$. For simplicity and convenience of presentation, we will denote $W^s \cdot g^{W^s}$ as $\Delta$ in the following.

**Algorithm 6** DistanceCompute.

```
Input:
    $X_e$: the ciphertext of a querier's location $X$;
    $Y_e$: the ciphertext of a publisher's location $Y$;
Output:
    $dis_e$: the ciphertext of the distance between $X$ and $Y$;
Obtain $m' = m \cdot W^s$ and $x' = x + W^s$ from the $Y_e^I$ ;
Compute $K'_Y = m' \cdot g^{x'} \bmod n$;
Compute $dis_e = \sqrt{\sum_{i=1}^{3} (x_{ei} - y_{ei} \cdot K'_Y)^2}$;
    return $dis_e$;
```

From Algorithm 6 and Equation (1), we know that for $i = 1, 2, 3$,

$$\begin{aligned} & x_{ei} - y_{ei} \cdot K'_Y \\ & = x_i \cdot \Delta - y_i \cdot (mg^x)^{-1} \cdot K'_Y \\ & = (x_i - y_i) \cdot \Delta \end{aligned}$$

Thus,

$$dis_e = \sqrt{\sum_{i=1}^{3} (x_{ei} - y_{ei} \cdot K'_Y)^2} = \sqrt{\sum_{i=1}^{3} (x_i - y_i)^2} \cdot \Delta = dis \cdot \Delta \tag{11}$$

After executing Algorithm 6, the mobile cloud sends $dis_e$ to the querier. Finally, the querier can get the plain distance $dis$ by computing Equation (12).

$$dis = dis_e \cdot \Delta^{-1}. \tag{12}$$

During the execution, all that the mobile cloud processes is the ciphertext.

2. Location distance compare:

The querier wants to know whether the location distance is within a threshold value $\tau$. He/she encrypts the $\tau$ as $\tau_e$, using Equation (13):

$$\tau_e = \tau \cdot \Delta. \tag{13}$$

Then, the querier sends $X_e$ and $\tau_e$ to the mobile cloud. The mobile cloud executes Algorithm 7 to compute whether the distance between $X$ and $Y$ is less than $\tau$.

From Equation (11), we know that $dis_e = dis \cdot \Delta$. Since $\Delta$ is always positive, it will not change the compare result between $dis$ and $\tau$. Thus, the mobile cloud can give out the comparison results through Algorithm 7 directly.

**Algorithm 7** DistanceCompare.

```
Input:
    $X_e$: the ciphertext of a querier's location $X$;
    $Y_e$: the ciphertext of a publisher's location $Y$;
    $\tau_e$: the ciphered compare parameter $\tau$;
Output:
    $true$ or $false$;
    Execute $dis_e = DistanceCompute\left(X_e, Y_e\right)$;
    if $dis_e - \tau_e \leq 0$ then return $true$;
    end if
    return $false$;
```

## 6. Security Analysis

In our system, the publisher can authorize the queriers that he/she knows or not, such as his/her friends or someone who has similar interests. Hence, the queries may include attackers. If so, it is easy for the attacker to get a certain plaintext/ciphertext pair. Thus, our scheme has to be secure against the chosen plaintext attack. Next, we will prove it.

**Theorem 1.** *Our scheme is secure against CPA.*

**Proof of Theorem 1.** Assume an attacker obtains $Y = \{y_1, y_2, y_3\}$ and its ciphertext $Y_e$. From Equation (1), we get that:

$$mg^x = \frac{y_i}{y_{ei}}, i = 1, 2, 3. \tag{14}$$

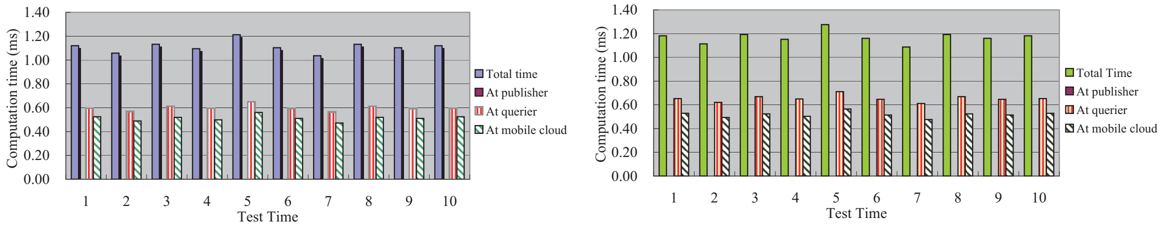
Assume that $B = mg^x$. It is easy to compute $B$. However, it is difficult to get the proper $\langle x, m \rangle$ from $B$. If $m$ is the power of $g$, it comes down obviously to the discrete logarithm problem. If $m$ is not the power of $g$, i.e., $m = m'g^{x'}$, where $m'$ is co-prime with $g$ and $x' \geq 0$, then $B = m'g^{x'+x}$. Even though the attacker can solve this equation, there are multiple solutions $x', x$ for $x' + x$. Let alone, it is even hard to solve $m, x' + x$ from $B$, which is as intractable as the discrete logarithm problem. As a consequence, the attacker cannot deduce the encryption key, as well as the secret parameter $W^s$ from $Y_e^I$. As described in Section 5, we know that secret parameter $W^s$ is the key to decrypt the query result. Thus, the attacker cannot decrypt extra confidential information apart from the already known $Y = \{y_1, y_2, y_3\}$. In conclusion, our scheme is secure against CPA. □

## 7. Experiment

As described in Section 2.1, the PLQP [5] is a suit of protocols supporting privacy-preserving LBS in mobile application. It has high efficiency and achieves fine-grained control by exerting the CP-ABE scheme, which is similar to our work. Thus, in this section, we will compare our scheme with the PLQP scheme [5] for evaluating the performance of our proposed scheme. The algorithms are implemented using the BigInteger library on a Windows 8.1 system with Intel CORE i7-4500U CPU@2.40 GHz and 8.00 G RAM. We have 10 tests in this experiment. Additionally, in each test, we use 1000 pairs of random locations for the publisher and the querier, respectively. We present the average results for each test in the following figures.

### *7.1. The Time Cost in Our Scheme*

Figure 5 shows the detailed time cost for once location distance compute and location distance compare, respectively. It is obvious that the time cost at the publisher is always zero, which meets the *Operate*algorithm in Section 5. The query processing work can be done with no need for the publisher's help.



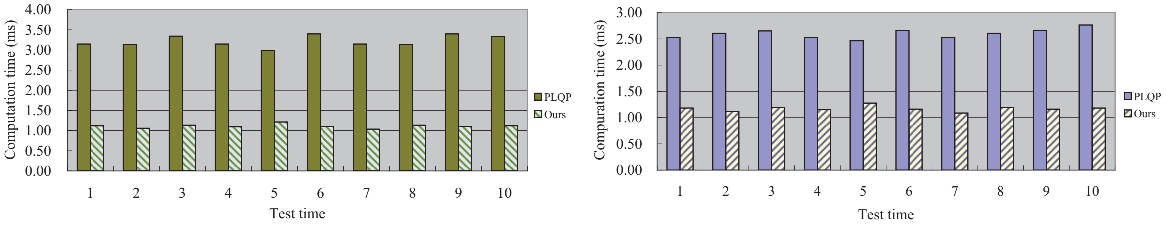
(a) The time cost of once location distance compute (b) The time cost of once location distance compare

**Figure 5.** The time cost in our scheme.

### 7.2. The Comparison of The Time Cost between Our Scheme and the PLQP Scheme

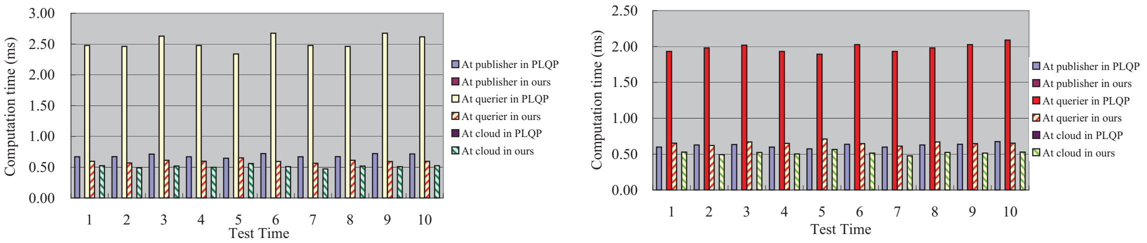
In Figure 6, we show the comparison results of the total time cost for the aforementioned two operations, respectively. It can be seen that our scheme is much more efficient than PLQP. Figure 7 shows the comparison results of the detailed time cost for the aforementioned two operations at the publisher, querier and mobile cloud, respectively. To be more clear, we also present the comparison results in Figures 8 and 9. From these figures, we get three points:

- At the querier, the time cost in our scheme is much less that that in PLQP.
- At the publisher, the time cost in our scheme is zero.
- At the mobile cloud, the time cost in PLQP is zero.



(a) Location distance compute (b) Location distance compare

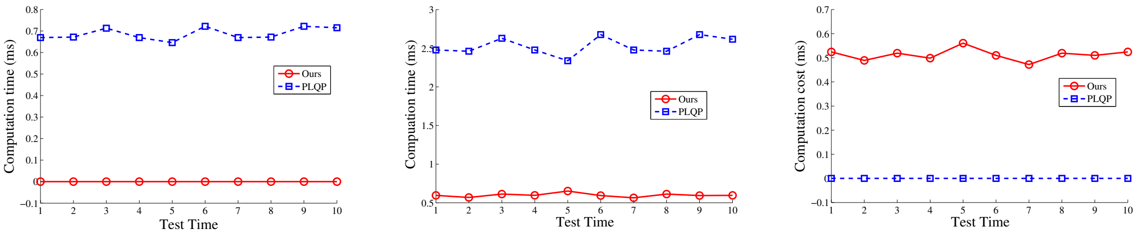
**Figure 6.** The comparison of the total time cost between our scheme and PLQP.



(a) Location distance compute (b) Location distance compare

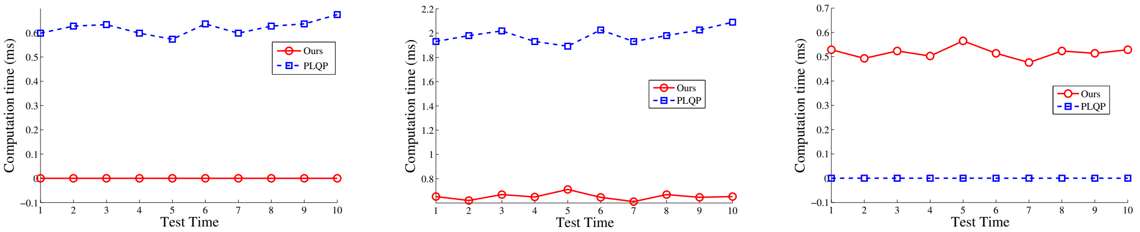
**Figure 7.** The comparison of each entity's time cost between our scheme and PLQP.

In sum, our scheme takes better advantage of the mobile cloud than PLQP and outperforms the PLQP scheme in terms of query efficiency. Numerical information about the computation cost is also shown in Table 2.



(a) At the publisher (b) At the querier (c) At the cloud

**Figure 8.** The time comparison of performing location distance compute between our scheme and PLQP.



(a) At publisher (b) At querier (c) At cloud

**Figure 9.** The time comparison of performing location distance compare between our scheme and PLQP.

**Table 2.** The computation cost comparison.

| Queries | Average Computation Time at the Querier (ms) | | Average Computation Time at the Publisher (ms) | | Average Computation Time at the Mobile Cloud (ms) | |
|---|---|---|---|---|---|---|
| | Ours | PLQP | Ours | PLQP | Ours | PLQP |
| Distance Compute | 0.59850795 | 2.52919789 | 0 | 0.68694300 | 0.51267655 | 0 |
| Distance Compare | 0.65277693 | 1.98040717 | 0 | 0.62046427 | 0.51697024 | 0 |

## 8. Conclusions

In this paper, we propose a privacy-preserving LBS scheme for mobile sensing data, by exerting the RSA algorithm and CP-ABE scheme. Our proposed scheme can support the mobile cloud to perform efficient and privacy-preserving queries of location distance compute and compare on encrypted locations. Moreover, due to the application of CP-ABE, our scheme achieves fine-grained control over the sensitive location data, where only authorized queriers, whose attributes satisfy the corresponding access tree, can decrypt the ciphered query results provided by the mobile cloud. As a consequence, both the publisher's and querier's location information are kept secret from the mobile cloud and unauthorized users. Finally, the security analysis proves that it is secure against CPA, and the performance evaluation demonstrates its efficiency with experiments compared with the efficient PLQP.

**Acknowledgments:** The work is supported by the National Natural Science Foundation of China under Grant No. 61472001 and No. U1405255, the project of Academic Leaders Funding of Anhui University and the Key Research and Development Project of Jiangsu Province under Grant No. BE2015136.

**Author Contributions:** Qingqing Xie proposed the idea of the research, conceived of, designed and performed the experiments. Liangmin Wang designed the structure and instructed Qingqing Xie to write the paper.

**Conflicts of Interest:** The authors declare no conflict of interest.

## Abbreviations

The following abbreviations are used in this manuscript:

| | |
|---|---|
| GPS | Global Positioning System |
| LBS | Location-based service |
| HE | Homomorphic encryption |
| ABE | Attribute-based encryption |
| CP-ABE | Ciphertext-policy attribute-based encryption |
| KP-ABE | Key-policy attribute-based encryption |
| AES | Advanced encryption standard |
| DES | Data encryption standard |
| PLQP | Privacy-preserving location query protocol |
| DLP | Discrete logarithm problem |
| PPT | Probabilistic polynomial-time |
| DDH | Decisional Diffie–Hellman |
| DBDH | Decisional Bilinear Diffie–Hellman |
| CPA | Chosen plaintext attack |
| TA | Trusted authority |

## References

1. Kos, A.; Tomažič, S.; Umek, A. Evaluation of Smartphone Inertial Sensor Performance for Cross-Platform Mobile Applications. *Sensors* **2016**, *16*, 477.
2. Liu, Z.; Niu, X.; Lin, X.; Huang, T.; Wu, Y.; Li, H. A Task-Centric Cooperative Sensing Scheme for Mobile Crowdsourcing Systems. *Sensors* **2016**, *16*, 746.
3. Dinh, T.; Kim, Y. A Novel Location-Centric IoT-Cloud Based on-Street Car Parking Violation Management System in Smart Cities. *Sensors* **2016**, *16*, 810.
4. Xie, Q.; Wang, L. Efficient privacy-preserving processing scheme for location-based queries in mobile cloud. In Proceedings of the IEEE International Conference on Data Science in Cyberspace, Changsha, China, 13–16 June 2016.
5. Li, X.Y.; Jung, T. Search me if you can: Privacy-preserving location query service. In Proceedings of the IEEE INFOCOM, Turin, Italy, 14–19 April 2013; pp. 2760–2768.
6. Ahn, J.; Han, R. MyBlackBox: Blackbox Mobile Cloud Systems for Personalized Unusual Event Detection. *Sensors* **2016**, *16*, 753.
7. Zou, H.; Jiang, H.; Luo, Y.; Zhu, J.; Lu, X.; Xie, L. BlueDetect: An iBeacon-Enabled Scheme for Accurate and Energy-Efficient Indoor-Outdoor Detection and Seamless Location-Based Service. *Sensors* **2016**, *16*, 268.
8. Shao, J.; Lu, R.; Lin, X. FINE: A fine-grained privacy-preserving location-based service framework for mobile devices. In Proceedings of the IEEE INFOCOM, Toronto, ON, Canada, 27 April–2 May 2014; pp. 244–252.
9. Zhang, B.; Liu, C.H.; Lu, J.; Song, Z.; Ren, Z.; Ma, J.; Wang, W. Privacy-Preserving QoI-Aware Participant Coordination for Mobile Crowdsourcing. *Comput. Netw.* **2016**, *101*, 29–41.
10. Shao, J.; Lu, R.; Lin, X. Fine-grained data sharing in cloud computing for mobile devices. In Proceedings of the IEEE INFOCOM, Hong Kong, China, 26–30 April 2015; pp. 2677–2685.
11. Gorbunov, S.; Vaikuntanathan, V.; Wee, H. Attribute-based encryption for circuits. *J. ACM (JACM)* **2015**, *62*, 45.
12. Zhang, K.; Gong, J.; Tang, S.; Chen, J.; Li, X.; Qian, H.; Cao, Z. Practical and Efficient Attribute-Based Encryption with Constant-Size Ciphertexts in Outsourced Verifiable Computation. In Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security, Xi'an, China, 30 May–3 June 2016; pp. 269–279.
13. Jung, T.; Li, X.Y.; Wan, Z.; Wan, M. Rebuttal to “Comments on ’Control Cloud Data Access Privilege and Anonymity With Fully Anonymous Attribute-Based Encryption’”. *IEEE Trans. Inf. Forensics Secur.* **2016**, *11*, 868.
14. Kitagawa, T.; Kojima, H.; Attrapadung, N.; Imai, H. Efficient and Fully Secure Forward Secure Ciphertext-Policy Attribute-Based Encryption. In *Information Security*; Springer: New York, NY, USA, 2015; pp. 87–99.

15. Doröz, Y.; Sunar, B. *Flattening NTRU for Evaluation Key Free Homomorphic Encryption*; Technical Report, Report 2016/315; Cryptology ePrint Archive: Worcester, MA, USA, 31 May 2016.
16. Gentry, C.; Groth, J.; Ishai, Y.; Peikert, C.; Sahai, A.; Smith, A. Using fully homomorphic hybrid encryption to minimize non-interative zero-knowledge proofs. *J. Cryptol.* **2015**, *28*, 820–843.
17. Ducas, L.; Micciancio, D. FHEW: Bootstrapping homomorphic encryption in less than a second. In *Advances in Cryptology–EUROCRYPT 2015*; Springer: New York, NY, USA, 2015; pp. 617–640.
18. Wang, W.; Hu, Y.; Chen, L.; Huang, X.; Sunar, B. Exploring the feasibility of fully homomorphic encryption. *IEEE Trans. Comput.* **2015**, *64*, 698–706.
19. Paillier, P. Public-key cryptosystems based on composite degree residuosity classes. In *Advances in Cryptology-EUROCRYPT'99*; Springer: New York, NY, USA, 1999; pp. 223–238.
20. Nishide, T.; Sakurai, K. Distributed paillier cryptosystem without trusted dealer. In *Information Security Applications*; Springer: New York, NY, USA, 2010; pp. 44–60.
21. Pan, M.; Sun, J.; Fang, Y. Purging the back-room dealing: Secure spectrum auction leveraging paillier cryptosystem. *IEEE J. Sel. Areas Commun.* **2011**, *29*, 866–876.
22. Xu, D.; Wang, R.; Shi, Y.Q. Data hiding in encrypted H. 264/AVC video streams by codeword substitution. *IEEE Trans. Inf. Forensics Sec.* **2014**, *9*, 596–606.
23. Dou, Y.; Zeng, K.C.; Yang, Y. Poster: Privacy-Preserving Server-Driven Dynamic Spectrum Access System. In Proceedings of the 21st Annual International Conference on Mobile Computing and Networking, Paris, France, 7–11 September 2015; pp. 218–220.
24. Kido, H.; Yanagisawa, Y.; Satoh, T. Protection of location privacy using dummies for location-based services. In Proceedings of the 21st International Conference on Data Engineering Workshops, Tokyo, Japan, 5–8 April 2005; p. 1248.
25. Duckham, M.; Kulik, L. A formal model of obfuscation and negotiation for location privacy. In *Pervasive Computing*; Springer: New York, NY, USA, 2005; pp. 152–170.
26. Chow, C.Y.; Mokbel, M.F.; Liu, X. A peer-to-peer spatial cloaking algorithm for anonymous location-based service. In Proceedings of the 14th annual ACM International Symposium on Advances in Geographic Information Systems, Arlington, VA, USA, 5–11 November 2006; pp. 171–178.
27. Mokbel, M.F. Towards privacy-aware location-based database servers. In Proceedings of the 22nd International Conference on Data Engineering Workshops, Atlanta, GA, USA, 3–7 April 2006; p. 93.
28. Bamba, B.; Liu, L.; Pesti, P.; Wang, T. Supporting anonymous location queries in mobile environments with privacygrid. In Proceedings of the 17th International Conference on World Wide Web, Beijing, China, 21–25 April 2008; pp. 237–246.
29. Gedik, B.; Liu, L. Protecting location privacy with personalized k-anonymity: Architecture and algorithms. *IEEE Trans. Mob. Comput.* **2008**, *7*, 1–18.
30. Shankar, P.; Ganapathy, V.; Iftode, L. Privately querying location-based services with SybilQuery. In Proceedings of the 11th International Conference on Ubiquitous Computing, Orlando, FL, USA, 30 September–3 October 2009; pp. 31–40.
31. Xue, M.; Kalnis, P.; Pung, H.K. Location diversity: Enhanced privacy protection in location based services. In *Location and Context Awareness*; Springer: New York, NY, USA, 2009; pp. 70–87.
32. Catalano, D.; Gennaro, R.; Howgrave-Graham, N.; Nguyen, P.Q. Paillier's cryptosystem revisited. In Proceedings of the 8th ACM Conference on Computer and Communications Security, Philadelphia, PA, USA, 5–8 November 2001; pp. 206–214.
33. San, I.; At, N.; Yakut, I.; Polat, H. Efficient paillier cryptoprocessor for privacy-preserving data mining. *Secur. Commun. Netw.* **2016**, *9*, 1535–1546.
34. Jung, T.; Li, X.Y.; Wan, Z.; Wan, M. Control Cloud Data Access Privilege and Anonymity With Fully Anonymous Attribute-Based Encryption. *IEEE Trans. Inf. Forensics Secur.* **2015**, *10*, 190–199.
35. Han, J.; Susilo, W.; Mu, Y.; Zhou, J.; Au, M.H.A. Improving privacy and security in decentralized ciphertext-policy attribute-based encryption. *IEEE Trans. Inf. Forensics Secur.* **2015**, *10*, 665–678.
36. Qin, B.; Deng, R.H.; Liu, S.; Ma, S. Attribute-Based Encryption with Efficient Verifiable Outsourced Decryption. *IEEE Trans. Inf. Forensics Secur.* **2015**, *10*, 1384–1393.
37. Xu, J.; Wen, Q.; Li, W.; Jin, Z. Circuit Ciphertext-Policy Attribute-Based Hybrid Encryption with Verifiable Delegation in Cloud Computing. *IEEE Trans. Parallel Distrib. Syst.* **2016**, *27*, 119–129.

38. Sahai, A.; Waters, B. Fuzzy identity-based encryption. In *Advances in Cryptology–EUROCRYPT 2005*; Springer: Berlin/Heidelberg, Germany, 2005; Volume 3494, pp. 457–473.
39. Goyal, V.; Pandey, O.; Sahai, A.; Waters, B. Attribute-based encryption for fine-grained access control of encrypted data. In Proceedings of the 13th ACM Conference Computer and Communications Security, Alexandria, VA, USA, 30 October–3 November 2006; pp. 89–98.
40. Bethencourt, J.; Sahai, A.; Waters, B. Ciphertext-policy attribute-based encryption. In Proceedings of the IEEE Symposium on Security and Privacy, Oakland, CA, USA, 20–23 May 2007; pp. 321–334.
41. Rivest, R.L.; Shamir, A.; Adleman, L. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM* **1978**, *21*, 120–126.
42. Liu, Y.; Han, J.; Wang, J. Rumor riding: Anonymizing unstructured peer-to-peer systems. *IEEE Trans. Parall. Distrib. Syst.* **2011**, *22*, 464–475.