Personal and Ubiquitous Computing (2018) 22:453–469
https://doi.org/10.1007/s00779-018-1124-7

ORIGINAL ARTICLE

Protecting query privacy with differentially private *k*-anonymity in location-based services

Jinbao Wang[1] · Zhipeng Cai[2] · Yingshu Li[2] · Donghua Yang[1] · Ji Li[2] · Hong Gao[1]

Received: 12 July 2017 / Accepted: 30 October 2017 / Published online: 9 March 2018

Abstract

Nowadays, location-based services (LBS) are facilitating people in daily life through answering LBS queries. However, privacy issues including *location privacy* and *query privacy* arise at the same time. Existing works for protecting *query privacy* either work on trusted servers or fail to provide sufficient privacy guarantee. This paper combines the concepts of differential privacy and k-anonymity to propose the notion of differentially private k-anonymity (DPkA) for *query privacy* in LBS. We recognize the sufficient and necessary condition for the availability of 0-DPkA and present how to achieve it. For cases where 0-DPkA is not achievable, we propose an algorithm to achieve ϵ-DPkA with minimized ϵ. Extensive simulations are conducted to validate the proposed mechanisms based on real-life datasets and synthetic data distributions.

Keywords k-Anonymity · Differential privacy · Query privacy · Location-based service

1 Introduction

Mobile devices equipped with positioning modules [32] have leveraged significant feasibility in our daily life through location-based service (LBS). By submitting a LBS query attached with one's location and a specified query interest, people may obtain points of interests (POI), such as bars or restaurants, nearby to facilitate daily activities. While enjoying the convenience brought by an untrusted LBS provider, users take the risk of privacy concerns from two aspects denoted *location privacy* and *query privacy* [26], since the disclosure of a user's location or query interests incurs potential damage to personal privacy and even to individual safety.

Similar to that of other aspects such as social network [17], to the privacy issue in LBS have lots of efforts been devoted for both *location privacy* and *query privacy*. For *location privacy*, both server-based and client-based solutions have been proposed. At the same time, *query privacy* protection is mainly protected by server-based solution, which includes a third-party trusted server to hide the querier among $k-1$ other users with cloaking technique [11, 12]. These server-based cloaking solutions suffer from potential single point of failure and computation bottleneck at the trusted server. Existing client-based solutions [30] also fail to provide sufficient privacy guarantee, for instance meaningful k-anonymity against the prior probability of each query interest held by LBS provider, to protect users' query interests.

In this paper, we combine the concepts of differential privacy and k-anonymity. Figure 1 depicts the work flow of our *query privacy* protection method. The module denoted differentially private k-anonymizer lies in users' mobile devices, and it stores k-sets of query interests in k-set pool. Given a query $Q(loc, I[j])$ with query interest $I[j]$, selection control picks a k-set $s^k = \{I[j], I[j_1], ..., I[j_{k-1}]\}$ containing the specified query interest $I[j]$ from k-set pool. Here, $I[j]$ must exists in s^k for the query utility requirement. The picked k-set contains $k-1$

✉ Donghua Yang
yang.dh@hit.edu.cn

Jinbao Wang
wangjinbao@hit.edu.cn

Zhipeng Cai
zcai@gsu.edu

Yingshu Li
yili@gsu.edu

Ji Li
jli30@student.gsu.edu

Hong Gao
honggao@hit.edu.cn

1 Harbin Institute of Technology, Harbin, China

2 Georgia State University, Atlanta, GA 30303, USA

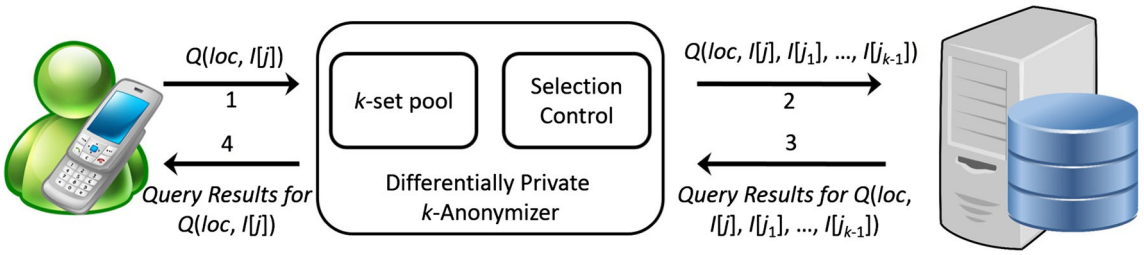


Fig. 1 Work Flow of Differentially Private k-Anonymizer

dummy query interests $I[j_1]$, ..., $I[j_{k-1}]$, and differentially private k-anonymizer submits them together with the actual query interest $I[j]$ to the LBS provider. Here, I is the global set of query interests, and $I[j]$ denotes the jth query interest in I. We aim to make the LBS provider unable to distinguish the actual query interest $I[j]$ and the $k-1$ dummies $I[j_1]$, ..., $I[j_{k-1}]$ through probabilistic inference, and differential privacy is adopted to achieve this goal. To this end, we define a notion for *query privacy* in LBS denoted Differentially Private ***k***-anonymity (DPkA), and study protecting *query privacy* based on DPkA using a client-based solution. Intuitively, DPkA is achieved if the posterior probability of any two query interests among the k submitted is close enough. Here, the term "close enough" is controlled by a privacy budget $\epsilon \geqslant 0$, and it is independent with the prior probability of each query interest, which follows the principle of differential privacy [13] and is different from the concept of geo-indistinguishability [4]. Such definition makes two submitted query interests difficult to distinguish even they have quite different prior probabilities. After introducing our *query privacy* notion, we recognize the sufficient and necessary condition for the availability of 0-DPkA, in which the posterior probability of any query interest from the k submitted is identical to $\frac{1}{k}$. We present how to achieve this perfect version of DPkA. Then, we formulate the problem of achieving the optimal DPkA with the maximized privacy as a non-linear programming problem with exponential scale of variables. It is not practical to achieve the optimal *query privacy* through this formulation. To overcome this issue, we present the problem of achieving ϵ-transformed 0-DPkA, which is achievable iff ϵ-DPkA is also achievable given a privacy budget ϵ and the prior probability of query interest $prob(.)$. We formulate the problem of achieving ϵ-transformed 0-DPkA as a linear programming problem with linear scale variables, so we can solve the problem efficiently. By solving the problem of ϵ-transformed 0-DPkA, we conduct a solution to the problem of achieving ϵ-DPkA with minimized privacy budget ϵ.

This paper makes the following contributions:

- A meaningful notion, differentially private k-anonymity (DPkA), is proposed for *query privacy* in LBS. It combines differential privacy [13] and k-anonymity to fit the privacy and utility requirements of query interests in LBS.
- We recognize a sufficient and necessary condition for the availability of 0-DPkA, under which the posterior probability of the k -submitted query interests is identical. An algorithm for building a mechanism for achieving 0-DPkA is given.
- For general cases, we formulate the problem of achieving the optimal DPkA in LBS as a non-linear programming problem with exponential variables. We then propose the ϵ-transformed 0-DPkA problem, which is equivalent to the problem of achieving the optimal DPkA but only involves linear scale of variables in the linear programming formulation. By solving the corresponding ϵ-transformed 0-DPkA problem, we propose a mechanism for achieving ϵ-DPkA with minimized privacy budget ϵ.
- We conduct a simulation based on real-life datasets from *OpenStreetMap* project [10], and through extensive simulation, we validate the effectiveness and efficiency of our proposed *query privacy* -protecting mechanism.

We outline the organization for the rest of this paper as follows. Section 2 introduces necessary preliminary on differential privacy and k-anonymity. Section 3 presents the notion of DPkA. The sufficient and necessary condition for 0-DPkA is given in Section 4, introduces how to build a mechanism for achieving the optimal DPkA in general cases. Our proposed mechanisms are evaluated with real-life datasets and synthetic data distributions in Section 5. Section 6 presents related works in the literature. The paper is concluded in Section 7.

2 Preliminary

This section introduces necessary preliminary on differential privacy, geo-indistinguishability and k-anonymity which are employed for preserving privacy in LBS. After that, we depict the adversary model in this paper, and notions used in the rest of this paper are also listed in this section.

2.1 Differential privacy

Differential privacy is a notion defined in the literature of statistic databases, and it is designed to prevent leakage of any individual's information in the process of answering aggregation queries in statistical databases. To satisfy the notion of differential privacy, a random algorithm should return query results with little difference in term of probabilistic distribution for two databases differing with just one insertion, deletion, or modification. In other words, a single change in a database brings minor modification to query results under the constraints of differential privacy. The definition of differential privacy is as below.

Definition 1 (Differential privacy) Given $\epsilon \geqslant 0$, a randomized algorithm $\mathcal{A}$ satisfies ϵ-differential privacy if for all neighboring databases D and D', $Prob(\mathcal{A}(D) \in S) \leqslant e^{\epsilon} \times Prob(\mathcal{A}(D') \in S)$. Here, $S \subseteq Range(\mathcal{A})$. Any pair of neighboring databases D and D' satisfies one of the following conditions: (1) (for unbounded differential privacy) D can be transformed to D' with exact one insertion or deletion and (2) (for bounded differential privacy) D can be transformed to D' with exact one modification.

The bounded differential privacy prevents distinguishing two datasets with the same size differing with exact one tuple, while the unbounded differential privacy prevents distinguishing two datasets which are the same except that one of them owns exact one additional tuple.

In the area of LBS, geo-indistinguishability is proposed to protect the $location\ privacy$. It shares the same idea to make a randomized algorithm produce similar outputs for nearby locations in term of probabilistic distribution. Intuitively, geo-indistinguishability guarantees that the ratio of posterior probability for two nearby locations, with regard to the result of the randomized algorithm, is similar to the ratio of their prior probability. In the following, we introduce a formal definition of geo-indistinguishability.

Definition 2 (Geo-indistinguishability) Given $\epsilon \geqslant 0$, a randomized algorithm $\mathcal{A}$ satisfies ϵ-geo-indistinguishability iff for any $r > 0$, any pair of location x and x' that $d(x, x') \leqslant r$ and any $S \subseteq Range(\mathcal{A})$, the following condition holds.

$$\frac{Prob(x|S)}{Prob(x'|S)} \leqslant e^{\epsilon \times r} \times \frac{Prob(x)}{Prob(x')}$$

The notion of geo-indistinguishability guarantees that a randomized algorithm does not bring significant leakage, and for two locations nearby, the ratio of posterior probability does not grow more than $e^{\epsilon \times r}$ times compared to the ratio of prior probability. However, the notion never bounds the ratio of posterior probability for two locations, since the ratio of their prior probability which could be very large is always involved. In Section 3, we present a mirror variant for $query\ privacy$ -denoted query-indistinguishability and propose a mechanism to achieve it. This provides a lower bound of privacy we can achieve in this paper.

2.2 *k*-Anonymity

k-Anonymity, formulated by Latanya Sweeney in 2002 [34], is proposed to guarantee that the protected target can not be distinguished from $k-1$ objects. This property is well adopted for preserving $location\ privacy$ and $query\ privacy$ in server-based solutions and client-based solutions.

The server-based solutions such as [5, 12, 38] include a third-party but trusted server, and LBS queries are first sent to the trusted server. After receiving LBS queries, the trusted server hides the user in $k-1$ users by generalization of LBS queries, and the attackers cannot recognize what (or where) does the user query.

The client-based solutions including [28–30] run at users' devices and generate $k-1$ dummies in a local manner, in which process certain side information is adopted, for instance, the prior probability of each queried location or interest.

Though the server-based solutions are effective, the trusted server may become a single failure if it is hacked by attackers and it is the computation bottleneck to incur long latency to LBS queries. At the same time, the existing client-based solutions provide specious k-anonymity, since the attackers may violate the principle of k-anonymity through rerunning of the algorithms or probability inference for each of the k results.

2.3 Adversary model

In this paper, we take the untrusted LBS server as the adversary, and we aim to prevent it from inferencing what does a user query. We adopt the common setting from the existing works, such as [4, 28, 30] , etc., that the adversary (1) sees k query interests one of which is true; (2) hold side information including the prior probability of each query interest, i.e., $Prob(.)$; and (3) is aware of the mechanism we adopt to generate the $k-1$ dummy query interests. The prior probability of each query interest is known to us at the same time, since we can adopt the solution presented in [28] to obtain such side information held by the adversary.

2.4 Notation and description

Here, we list notations and their description which are used in the rest of this paper as shown in Table 1.

Table 1 Summary of notations and description

Notation	Description
I	the set of query interests
$I[j]$	the jth query interest in I
k	the number of query interests reported
s_I^k	the k-set of query interests
F_I^k	the covering family of k-set of query interests
$F_I^k[i]$	the ith k-set of query interest in F_I^k
$prob(I[j])$	the prior probability of query interest $I[j]$
$prob(I[j]\|s_I^k)$	the posterior probability of $I[j]$ given s_I^k
P_I^k	the probability assignment matrix
$P_I^k[i][j]$	the probability of reporting $F_I^k[i]$ given $I[j]$
ϵ	the privacy budget in DP or DPkA

3 Differentially private *k*-anonymity in LBS

In this section, we first present a mirror variant of the aforementioned denoted query-indistinguishability for *query privacy* in LBS. Then, we give a mechanism to achieve the notion of query-indistinguishability and analyze the privacy level it guarantees. This privacy level is a lower bound of *query privacy* that we achieve in this paper. In the end of this section, we present the notion of differentially private k-anonymity, and mechanisms to achieve this notion will be proposed in Section 4.

3.1 *Query-indistinguishable k*-anonymity

The notion of query-indistinguishable k-anonymity requires a randomized mechanism not to provide significant improvement to the adversary's inference after the result of the randomized mechanism is seen. This underlying idea is a straightforward mapping from *query privacy*. The formal definition of query-indistinguishable k-anonymity is given as follows.

Definition 3 (Query-indistinguishable k-anonymity) Given $\epsilon \geqslant 0$, a randomized mechanism $\mathcal{A}$ satisfies ϵ-*query-indistinguishable* k-anonymity iff for any query interest $I[i]$, any result $s_I^k = \{I[i], I[i_1], ..., I[i_{k-1}]\} \in range(\mathcal{A}(I[i]))$, the following condition holds for any $I[j]$, $I[j'] \in s_I^k$:

$$\frac{Prob(I[j]|s_I^k)}{Prob(I[j']|s_I^k)} \leqslant e^{\epsilon} \times \frac{Prob(I[j])}{Prob(I[j'])}$$

Next, we present a randomized mechanism named *k-duplication* which achieve ϵ-*query-indistinguishable* k-anonymity. The basic idea of *k-duplication* is to generate a set of k-set S^k with $|I|$ elements. For each query interest $I[i] \in I$, there are exact k elements from S^k containing $I[i]$. At the same time, each k-set in S^k contains k different elements. When a user u queries with $I[i]$, *k-duplication* chooses one element from the k elements containing $I[i]$ from S^k uniformly. The detailed process of *k-duplication* is given in Algorithm 1 as follows.

Algorithm 1 *k-Duplication*

Input: query interest $I[i]$, query interest set I, integer $k < |I|$
Output: k-set s^k

```
1 S = k-DuplicationCandidate(I, k);
2 Pool = {s|s ∈ S ∧ I[i] ∈ s};
3 uniformly sample one element from Pool as s^k;
4 return s^k;
```

Algorithm 2 *k-DuplicationCandidate*

Input: query interest set I, integer $k < |I|$
Output: a set S of k-sets

```
1  S = φ;
2  for d=0 to |I| − 1 do
3      S_d = φ;
4  cusor = 0;
5  for I[j] ∈ I;
6  do
7      id = cusor % |I|;
8      S_id = S_id ∪ {I[j]};
9      cusor + +;
10 for d=0 to |I| − 1 do
11     S = S ∪ S_d;
12 return S;
```

Algorithm 2 generates candidates of k-set to report to LBS provider for all the query interests. Lines 1–3 initiate $|I|$ empty query interest sets. Lines 4–9 first create k duplicates for each query interest, then put all the duplicates into $|I|$ query interest sets in a round-robin manner. Finally, the union of all the abovementioned query interest sets is returned (lines 10–12). Since $k < |I|$ and there are $k \times |I|$ duplicates, Algorithm 2 returns a set S of I query interest sets, each of which contains k different query interests. What's more, each query interest appears in exact k query interest sets in S. Here, we assume that I has been sorted using $O(|I| \log |I|)$[22] computation time.

Algorithm 1 first invokes Algorithm 2 to obtain k-sets of query interests (line 1). Then, it retrieves all the elements in S which contain i, and put them into candidate set $Pool$ (line 2). Finally, Algorithm 1 uniformly samples one element from $Pool$ as the final result (lines 3–4).

Theorem 1 *The randomized mechanism of Algorithm 1 achieves 0-query-indistinguishable k-anonymity.*

Proof Recall that Algorithm 2 generates $|I|$ candidate query interest sets, and each of which consists of k different query interests. At the same time, each query interest $I[i] \in I$ appears in exact k elements from the candidate returned by Algorithm 2. Suppose the randomized mechanism proposed in Algorithm 1 returns $S = \{I[i_1], ..., I[i_k]\}$. For any $1 \leqslant j \leqslant k$, denote S^j as the set of k-sets generated in Algorithm 2 containing query interest $I[j]$. Due to the random sampling in Algorithm 1, $Prob(S|I[j]) = \frac{1}{k}$ (as stated in line 3). Similarly, we can get $Prob(S|I[j']) = \frac{1}{k}$ for any $I[j'] \in S$. Thus, for any $I[j]$ and $I[j']$ from the result S,

$$\frac{Prob(I[j]|S)}{Prob(I[j']|S)} = \frac{Prob(I[j] \wedge S)}{Prob(I[j'] \wedge S)}$$
$$= \frac{Prob(S|I[j]) \times Prob(I[j])}{Prob(S|I[j']) \times Prob(I[j'])}$$
$$= \frac{Prob(I[j])}{Prob(I[j'])} \leqslant e^0 \times \frac{Prob(I[j])}{Prob(I[j'])}$$

By the inequation above, we conclude that the randomized mechanism of Algorithm 1 achieves 0-*query-indistinguishable* k-anonymity. □

3.2 The notion of differentially private *k*-anonymity

The notion of query-indistinguishable k-anonymity ensures that the LBS provider cannot improve posterior probability of each query interest after is bounded by the product of a constant (determined by ϵ) and the ratio of the two query interests' prior probability. It is still easy to distinguish two query interests (recognize the true one with large probability) in a LBS query if they have prior probability with large difference, even query-indistinguishable k-anonymity is achieved.

To eliminate the side effect of prior probability, we define the notion of differentially private k-anonymity (DPkA) by removing the ratio term related to prior probability of query interests, and we get the following definition of DPkA.

Definition 4 (Differentially private k-anonymity) Given $\epsilon \geqslant 0$, a randomized mechanism $\mathcal{A}$ satisfies ϵ-DPkA iff for any query interest $I[i]$, any result $s_I^k = \{I[i], I[i_1], ..., I[i_{k-1}]\} \in Range(\mathcal{A}(I[i]))$, the following condition holds for any $I[j], I[j'] \in s_I^k$.

$$\frac{Prob(I[j]|s_I^k)}{Prob(I[j']|s_I^k)} = \frac{prob(I[j]) \times prob(s_I^k|I[j])}{prob(I[j']) \times prob(s_I^k|I[j'])} \leqslant e^\epsilon$$

The notion of DPkA provides more strict privacy constraint than that of query-indistinguishable k-anonymity, since it makes two query interests hard to distinguish even that they have prior probability with large difference. The randomized mechanism in Algorithm 1 no longer satisfies 0-DPkA. Instead, it provides a lower bound of privacy for the notion of DPkA as stated in the following theorem.

Theorem 2 *The randomized mechanism in Algorithm 1 achieves* $\max_{I[j],I[j']\in I}\{\ln \frac{Prob(I[j])}{Prob(I[j'])}\}$*-DPkA.*

Proof Since the randomized mechanism in Algorithm 1 achieves 0-*query-indistinguishable* k-anonymity as stated in Theorem 1, given the reported query interests as S, for any $I[j], I[j'] \in S$, we have

$$\frac{Prob(I[j]|S)}{Prob(I[j']|S)} \leqslant e^0 \times \frac{Prob(I[j])}{Prob(I[j'])} \leqslant e^{\ln \frac{Prob(I[j])}{Prob(I[j'])}}.$$

By traversing all the possible $I[j]$ and $I[j']$, we get that

$$\frac{Prob(I[j]|S)}{Prob(I[j']|S)} \leqslant e^{\max_{I[j],I[j']\in I}\{\ln \frac{Prob(I[j])}{Prob(I[j'])}\}}.$$

Thus, we conclude the randomized mechanism in Algorithm 1 achieves $\max_{I[j],I[j']\in I}\{\ln \frac{Prob(I[j])}{Prob(I[j'])}\}$-DP$k$A. □

Though ineffective, the randomized mechanism in Algorithm 1 provides a lower bound for the privacy we can achieve in term of our proposed notion of DPkA. That means we can achieve ϵ-DPkA with the privacy budget no larger than $\epsilon = \max_{I[j],I[j']\in I}\{\ln \frac{Prob(I[j])}{Prob(I[j'])}\}$. In fact, it is still possible to achieve smaller ϵ for better privacy, and in the next section, we study how to approach this goal.

4 Mechanisms for differentially private *k*-anonymity

This section formulates the problem of achieving DPkA for *query privacy*, and a naïve solution using non-linear programming with exponential scale of variables is presented. Then, we propose mechanisms for achieving DPkA in an efficient manner. First, we deal with a special case in which the privacy budget ϵ can be reduced to 0. Then, a mechanism is designed to deal with the condition when the privacy budget ϵ cannot be reduced to 0, and our proposed mechanism is able to achieve the minimum privacy budget for best privacy.

4.1 Problem definition

Before introducing the formal definition of our problem, we first present the definition of k-set of query interests, covering family of k-set and probability assignment matrix which form the basis of our problem definition.

Definition 5 (k-set of query interests) Given a global set of query interests denoted I together with an integer k, a k-set of query interests from I denoted s_I^k is a set of k different

query interests from I. We denote $s_I^k[i]$ the ith query interest in s_I^k, and denote $I[i]$ the ith query interest from I.

Definition 6 (Covering family of k-set) Given a global set of query interests denoted I together with an integer k, a covering family of k-set denoted F_I^k is a set of k-sets of query interests, and each query interest from I exists in at least one element from F_I^k. We denote $F_I^k[i]$ the ith k-set from F_I^k.

Definition 7 (Probability assignment matrix) Given a global set of query interests denoted I together with an integer k and one of its covering family of k-set F_I^k, a probability assignment matrix P_I^k based on F_I^k is a $|F_I^k| \times |I|$ matrix where each row corresponds to a k-set $s_I^k \in F_I^k$ and each column corresponds to a query interest from I. $P_I^k[i][j] = prob(F_I^k[i]|I[j])$ is the probability of reporting $F_I^k[i]$ given $I[j]$ satisfying that:

(1) for any $1 \leqslant i \leqslant |F_I^k|$ and $1 \leqslant j \leqslant |I|$, if $I[j] \in F_I^k[i]$ then $P[i][j] \geqslant 0$, otherwise $P[i][j] = 0$;
(2) for any $1 \leqslant j \leqslant |I|$, $\sum_{1 \leqslant i \leqslant |F_I^k|} P[i][j] = 1$.

A covering family of k-sets F_I^k and its corresponding probability assignment matrix P_I^k determines a randomized mechanism for preserving $query\ privacy$ in LBS. Given the actual query interest as $I[j]$, the mechanism picks one k-set from $F_I^k[i]$, and the probability of picking $F_I^k[i]$ is $P_I^k[i][j]$. Here, if $P_I^k[i][j] = 0$, $F_I^k[i]$ is never picked for $I[j]$. The picked k-set of $query\ privacy$ is to be reported to LBS provider for response, and the actual query interest is hidden in the reported k-set $F_I^k[i]$.

Next, we formulate the problem of achieving the optimal DPkA. To solve our proposed problem, one should compute a covering family of k-sets and a probability assignment matrix based on it. After the computation, the minimized privacy budget ϵ should be achieved for best privacy.

Definition 8 (Problem of achieving the optimal differentially private k-anonymity) Given a global set of query interest as I, the prior probability of any query interests $I[i] \in I$ as $prob(I[i])$ and an integer k. Computing a randomized mechanism $\mathcal{M}(F_I^k, P_I^k)$ for each $s_I^k = F_I^k[i]$ and any query interest $I[j]$, $I[j'] \in s_I^k$, the following condition holds

$$\frac{prob(I[j]|s_I^k)}{prob(I[j']|s_I^k)} = \frac{prob(I[j]) \times P_I^k[i][j]}{prob(I[j']) \times P_I^k[i][j']} \leqslant e^{\epsilon}.$$

At the meanwhile, ϵ is minimized with the constraint that $\epsilon \geqslant 0$.

4.2 A naïve solution

Here, we present a naïve solution to the problem proposed in Definition 8 using a non-linear programming technique. Given the global query interest set I, an integer k, the naïve solution puts all the k-sets based on I into F_I^k, so there are $\binom{|I|}{k}$ k-sets in total. For any $s_I^k \in F_I^k$, the following condition holds if $I[j]$, $I[j'] \in s_I^k$.

$$\frac{prob(I[j]) \times P_I^k[i][j]}{prob(I[j']) \times P_I^k[i][j']} \leqslant e^{\epsilon}.$$

The above condition can be transformed to the following constraint:

$$\ln prob(I[j]) + \ln P_I^k[i][j] \leqslant \epsilon + \ln prob(I[j']) + \ln P_I^k[i][j']$$

In the above constraint, we denote $x_{i,j} = \ln P_I^k[i][j]$ and $d_{j,j'} = \ln prob(I[j]) - \ln prob(I[j'])$ then the constraint could be written as follows:

$$x_{i,j} - x_{i,j'} - \epsilon + d_{j,j'} \leqslant 0.$$

There are $\binom{|I|}{k} \times \binom{k}{2} \times 2$ constants of the above form, and since we can pick $\binom{|I|}{k}$ values for i and for each i , there are $\binom{k}{2}$ values of j and j' the relation of which are symmetric at the same time. What's more, each value of $x_{i,j} \leqslant 0$, and such condition brings $\binom{|I|}{k} \times k$ constraints. For the property of probability, we have $\sum_{1 \leqslant i \leqslant |F_I^k|} P_I^k[i][j] = 1$ for $1 \leqslant j \leqslant |I|$. Thus, we have the following constraints for $1 \leqslant j \leqslant |I|$.

$$\sum_{1 \leqslant i \leqslant |F_I^k|} e^{x_{i,j}} = 1$$

The number of such constraints is $|I|$. Finally, we have $\epsilon \geqslant 0$ as the last constraint.

We can formulate the non-linear programming formulas for our naïve solution as follows:

$$\begin{aligned}
&min \quad \epsilon \\
&s.t. \quad x_{i,j} - x_{i,j'} - \epsilon + d_{j,j'} \leqslant 0 \quad \text{for } i, j, I[j] \in F_I^k[i] \\
&\qquad x_{i,j} \leqslant 0 \quad \text{for all entries in } P_I^k \\
&\qquad \sum_{1 \leqslant i \leqslant |F_I^k|} e^{x_{i,j}} = 1 \quad \text{for } 1 \leqslant j \leqslant |I| \\
&\qquad \epsilon \geqslant 0
\end{aligned}$$

The naïve solution includes $\binom{|I|}{k} \times k + 1$ variables, each of which corresponds to ϵ or an entry in P_I^k, together with $\binom{|I|}{k} \times \binom{k}{2} \times 2 + \binom{|I|}{k} \times k + |I| + 1$ constrains in total. The non-linear programming problem is not easy to compute, due to the non-linear property together with the exponential scale of variables and constraints. In the rest of this section, we propose an efficient method to solve the problem of achieving the optimal DPkA for $query\ priacy$ in LBS applications.

4.3 Dealing with a special case: $\epsilon = 0$

Here, we first present a necessary condition for the availability of 0-DPkA. Then, a mechanism is given for achieving the notion of 0-DPkA under the assumption of the necessary condition we present. Finally, we claim that the necessary condition we present is in fact a sufficient and necessary condition for the availability of 0-DPkA.

4.3.1 A necessary condition

The following theorem illustrates a necessary condition for the availability of 0-DPkA. Intuitively, this necessary condition requires that the prior probability of the reported query interests are not too far away.

Lemma 1 *Given the global query interest set I, and for each $I[j] \in I$ the prior probability of $I[j]$ is $prob(I[j])$, if 0-DPkA can be achieved then the following condition holds*

$$\max_{I[j]\in I} prob(I[j]) \leqslant \frac{1}{k}.$$

Proof Suppose we achieve 0-DPkA with covering family of k-sets denoted F_I^k and probability assignment matrix P_I^k. Notice that the property of 0-DPkA guarantees the following equation for each i, j that $I[j] \in F_I^k[i]$.

$$prob(I[j]) \times P_I^k[i][j] = \frac{1}{k} \times prob(F_I^k[i]).$$

Furthermore, for each $I[j] \in I$, the following condition holds.

$$\begin{aligned} prob(I[j]) &= \sum_{I[j]\in F_I^k[i]} \{prob(I[j]) \times P_I^k[i][j]\} \\ &= \sum_{I[j]\in F_I^k[i]} \{\frac{1}{k} \times prob(F_I^k[i])\} \\ &\leqslant \sum_{F_I^k[i]\in F_I^k} \{\frac{1}{k} \times prob(F_I^k[i])\} = \frac{1}{k}. \end{aligned}$$

By traversing all the $I[j] \in I$, we can make the conclusion that $\max_{I[j]\in I} prob(I[j]) \leqslant \frac{1}{k}$ holds. □

4.3.2 Achieving 0-differentially private k-anonymity

Here, we present an algorithm to achieve 0-DPkA under the assumption that $\max_{I[j]\in I} prob(I[j]) \leqslant \frac{1}{k}$. Before introducing our proposed algorithm, we first provide an example to demonstrate the underlying idea.

Given $I = \{I[1], I[2], I[3], I[4], I[5]\}$ and the prior probability of each query interest as follow, $prob(I[1]) = 0.3$, $prob(I[2]) = 0.25$, $prob(I[3]) = 0.2$, $prob(I[4]) = 0.15$, $prob(I[1]) = 0.1$. We set $k = 2$ in this example. We are to compute F_I^k and P_I^k for this instance. Instead of starting from fixing F_I^k, we seek a feasible solution of P_I^k and F_I^k is determined in the process of fixing P_I^k. As show in Fig. 2, we first create five segments seg_i ($i = 1, 2, 3, 4, 5$). Denoting the length of seg_i as $L(seg_i)$, we set $L(seg_i) = prob(I[i])$ for each i. In the next step, we align these segments along $k = 2$ layers with length of $\frac{1}{k} = 0.5$ from the bottom to the top. The first layer takes up seg_1 and part of seg_2 denoted seg_{21}. The length of $Layer$ 1 is 0.5 and it takes 0.2 from seg_2. The rest of seg_2 denoted seg_{22} with length 0.05 is aligned with $Layer$ 2, then seg_3, seg_4 , and seg_5 are aligned to $Layer$ 2. The length of $Layer$ 2 is also 0.5. In the third step, we extend the vertical dash line through each ending point of segments, and these dash lines cut $Layer$ 1 and $Layer$ 2 into five collections, each of which is a k-set of query interests in F_I^k. Thus, F_I^k contains $\{I[1], I[2]\}$, $\{I[1], I[3]\}$, $\{I[1], I[4]\}$, $\{I[2], I[4]\}$, and $\{I[2], I[5]\}$. In this process, we align the segments one by another onto the k $Layer$, and the length of each segment is more than $\frac{1}{k}$, so we conclude that each vertical line will not intersect with two segments for the same query interest (ending points are not included in a segment). Finally, we show the calculation of P_I^k. For a k-set $F_I^k[i]$ in the above process, suppose a segment seg_i is included in $F_I^k[i]$. If seg_j is include entirely, $P_I^k[i][j] = 1$. Otherwise, if seg_j is partially included as seg_{jm}, then $P_I^k[i][j] = \frac{L(seg_{jm})}{L(seg_j)}$. In the instance shown in Fig. 2, $P_I^k[1][1] = \frac{1}{6}$ and $P_I^k[1][2] = \frac{1}{5}$ for $I[1]$, $I[2]$ and $F_I^k[1] = \{I[1], I[2]\}$. By now, we get F_I^k and P_I^k and they consist of a mechanism achieving 0-DPkA.

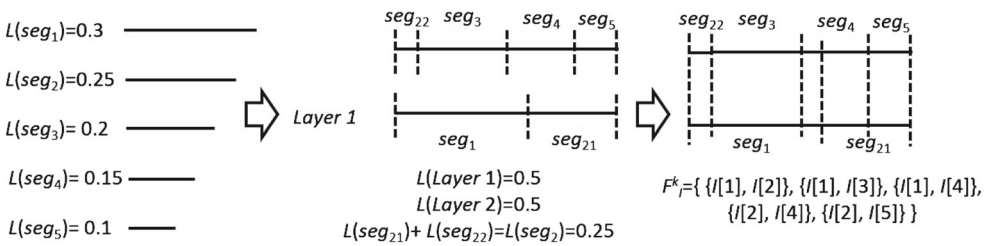


Fig. 2 An example of achieving 0-DPkA for an instance with $k = 2$, $I = \{I[1], I[2], I[3], I[4], I[5]\}$, and prior probability as $prob(I[1]) = 0.3$, $prob(I[2]) = 0.25$, $prob(I[3]) = 0.2$, $prob(I[4]) = 0.15$ and $prob(I[5]) = 0.1$

Algorithm 3 0-*DP-k-Anonymity*

Input: query interest set I, integer $k < |I|$, prior probability $prob(I[i])$ for each $I[i] \in I$
Output: covering family of k-set F_I^k, probability assignment matrix P_I^k

1 $F_I^k = \phi$;
2 initiate P_I^k;
3 $layer = 1$;
4 $cusor = 0$;
5 $finish = 0$;
6 **for** i=*1 to* k **do**
7 $list[i]$=new List();
8 **while** $finish < |I|$ **do**
9 **if** $\frac{1}{k} - cusor - length \leqslant 0$ **then**
10 $list[layer].add(< I[finish+1], \frac{1}{k} - cusor >)$;
11 $layer++$;
12 $cusor = 0$;
13 $length- = (\frac{1}{k} - cusor)$;
14 **if** $length == 0$ **then**
15 $finish++$;
16 $length = prob(I[finish+1])$;
17 **else**
18 $list[layer].add(< I[finish+1], length >)$;
19 $cusor+ = length$;
20 $finish++$;
21 $length = prob(I[finish+1])$;
22 $rowid = 0$;
23 $rest = \frac{1}{k}$;
24 **while** $rest > 0$ **do**
25 $s_I^k = \phi$;
26 **for** $i = 1$ *to* k **do**
27 $s_I^k = s_I^k \cup list[i].head().item()$;
28 $r = \arg_{1 \leqslant i \leqslant k} \min list[i].head().length()$;
29 $length = list[r].head().length()$;
30 **for** $i = 1$ *to* k **do**
31 $item = list[i].head().item()$;
32 $P_I^k[rowid][item.id()] = \frac{length}{prob(item)}$;
33 **if** $list[i].head().length() == length$ **then**
34 $list[i].remove(0)$;
35 **else**
36 reduce $list[i].head().length()$ by $length$;
37 $F_I^k = F_I^k \cup s_I^k$;
38 $rest- = length$;
39 $rowid++$;
40 return $< F_I^k, P_I^k >$;

The implementation of the above-described process is formally introduce in Algorithm 3. Algorithm 3 aligns I segments with length $prob(I[i])$ to k layers (lines 3–21), and then it computes F_I^k and P_I^k (lines 22–39). It traverses all the layers (lines 24–39) and during the traversing, Algorithm 3 seeks the next vertical dash line to generate k-set $s_I^k \in F_I^k$ (lines 25–27) and calculate each entry of P_I^k (lines 31–36).

The correctness of Algorithm 3 can be seen from two aspects. First, the process generates a number of k-sets, each of which consists of k different query interests. This is guaranteed by the assumption that the maximal prior probability of a query interest is no more then $\frac{1}{k}$. Second, we can get the posterior probability of two query interests from any $s_I^k \in F_I^k$ is identical due to the calculation of corresponding entries in P_I^k. Thus, Algorithm 3 returns a randomized mechanism $\mathcal{M}(F_I^k, P_I^k)$ achieving 0-DPkA under the assumption that $\max_{I[j] \in I} prob(I[j]) \leqslant \frac{1}{k}$.

Since each query interest incurs at most one dash line except the right end of each layer, the number of k-set in F_I^k is of $O(|I|)$. There are $O(k \times |I|)$ entries in P_I^k. So, the time complexity of Algorithm 3 is $O(k \times |I|)$.

4.3.3 A sufficient and necessary condition

As a straightforward consequence of Lemma 1 and the fact that Algorithm 3 achieves 0-DPkA under the assumption $\max_{I[j] \in I} prob(I[j]) \leqslant \frac{1}{k}$, we conclude the following theorem.

Theorem 3 *Given the query interest set I, and for each $I[j] \in I$, the prior probability of $I[j]$ is $prob(I[j])$, 0-DPkA can be achieved if the following condition holds:*

$$\max_{I[j] \in I} prob(I[j]) \leqslant \frac{1}{k}.$$

By here, we have studied the sufficient and necessary condition for the availability of 0-DPkA and Algorithm 3 is proposed for designing a randomized mechanism to achieve 0-DPkA. Next, we are to deal with general cases when 0-DPkA is not achievable.

4.4 Achieving the optimal differentially private *k*-anonymity

We first consider the decision version of the problem of achieving the optimal differentially private k-anonymity. The formal definition of the decision problem is given as follow.

Definition 9 (Decision problem of achieving the optimal differentially private k-anonymity) Given a global set of

query interest as I, the prior probability of any query interest $I[i] \in I$ as $prob(I[i])$, an integer k , and a privacy budget $\epsilon \geqslant 0$. Decide whether there is a randomized mechanism $\mathcal{M}(F_I^k, P_I^k)$ for each $s_I^k = F_I^k[i]$ and any query interest $I[j], I[j'] \in s_I^k$, the following condition holds:

$$\frac{prob(I[j]|s_I^k)}{prob(I[j']|s_I^k)} = \frac{prob(I[j]) \times P_I^k[i][j]}{prob(I[j']) \times P_I^k[i][j']} \leqslant e^{\epsilon}.$$

Next, we introduce a decision problem of ϵ-transformed 0-differentially private k-anonymity, which can be proved equivalent to the decision problem of achieving the optimal differentially private k-anonymity. To this end, we can solve the optimization problem of ϵ-transformed 0-differentially private k-anonymity, and then obtain a solution of the problem of achieving the optimal differentially private k-anonymity. The definition for the problem of ϵ-transformed 0-DPkA and its corresponding decision version as follows.

Definition 10 (Optimization problem of ϵ-transformed 0-DP k-anonymity) Given a global set of query interest as I, the prior probability of any query interest $I[i] \in I$ as $prob(I[i])$ and an integer k, compute $\epsilon_1, ..., \epsilon_{|I|} \geqslant 0$ satisfying that 0-DPkA is achievable for I if the prior probability of $I[j]$ is transformed to $\frac{prob(I[j])}{e^{\epsilon_j}}$. (Notice that the definition does not restraint the sum of transformed prior probability is 1.) At the same time, $\max_{1 \leqslant j \leqslant |I|}\{\epsilon_j\}$ is minimized.

Definition 11 (Decision problem of ϵ-transformed 0-DP k-anonymity) Given a global set of query interest as I, the prior probability of any query interest $I[i] \in I$ as $prob(I[i])$, an integer k and a privacy budget $\epsilon \geqslant 0$, decide whether there exists $0 \leqslant \epsilon_1, ..., \epsilon_{|I|} \leqslant \epsilon$ satisfying that 0-DPkA is achievable for I if the prior probability of $I[j]$ is transformed to $\frac{prob(I[j])}{e^{\epsilon_j}}$. (Notice that the definition does not restraint the sum of transformed prior probability is 1.)

The following theorem demonstrates the equality of the two decision problem introduced in Definitions 9 and 11.

Theorem 4 *Given an instance $(I, prob, k, \epsilon)$ for both the decision problem of achieving the optimal DPkA and ϵ-transformed 0-DP k-anonymity, the answers of the two problem are the same.*

Proof The proof includes two aspects. First, given an instance $(I, prob, k, \epsilon)$, we are to prove that if ϵ-dp k-anonymity is achievable, then ϵ-transformed 0-dp k-anonymity is also achievable.

Suppose ϵ-dp k-anonymity is achieved by the design of $\mathcal{M}(F_I^k, P_I^k)$. For each $F_I^k[i] \in F_I^k$, we denote $base_i = \arg_{I[m] \in F_I^k[i]} \min\{prob(I[m]) \times prob(F_I^k[i]|I[m])\}$. Thus, $base_i$ is the query interest with minimum value of $prob(.) \times prob(F_I^k|.)$. For each query interest $I[m'] \in F_i^k$, denote $\epsilon_{m',i} = \ln\{\frac{prob(I[m']) \times prob(F_I^k[i]|I[m'])}{prob(base_i) \times prob(F_I^k[i]|base_i)}\}$, or in other words, we have $e^{\epsilon_{m',i}} = \frac{prob(I[m']) \times prob(F_I^k[i]|I[m'])}{prob(base_i) \times prob(F_I^k[i]|base_i)}$. Now, we can build a solution to achieve ϵ-transformed 0-dp k-anonymity, and the built solution keeps F_I^k unchanged. For a query interest $I[x] \in I$ and each $I[x] \in F_I^k[i]$, we shrink the value of $prob(I[x]) \times prob(F_I^k[i]|)$ to $\frac{prob(I[x]) \times prob(F_I^k[i]|)}{e^{x,i}}$. After shrinking the target value of all the query interests in the k-set they appear, we get that $prob(I[x]) \times prob(F_I^k[i]|) = prob(base_i) \times prob(F_I^k|base_i)$ for all the query interests in the k-sets they appear. The P_I^k of the built solution changes with the shrinked value of each query interest, and we omit the details here since it does not related to the rest proof. By now, we can conclude that the transformed solution achieves 0-DPkA due to the Definition 4. To prove this solution satisfies ϵ-transformed 0-DPkA, we need to prove that there exists an ϵ_x satisfying that $\frac{prob(I[x])}{e^{\epsilon_x}} = \sum_{I[x] \in F_I^k[i]}\{\frac{prob(I[x]) \times prob(I[x]|F_I^k)}{e^{\epsilon_{x,i}}}\}$ and $\epsilon_x \leqslant \epsilon$. Since each $\epsilon_{x,i} \leqslant \epsilon$, we can get that

$$\frac{prob(I[x])}{e^{\epsilon_x}} = \sum_{I[x] \in F_I^k[i]} \{\frac{prob(I[x]) \times prob(I[x]|F_I^k)}{e^{\epsilon_{x,i}}}\}$$

$$\geqslant \sum_{I[x] \in F_I^k[i]} \{\frac{prob(I[x]) \times prob(I[x]|F_I^k)}{e^{\epsilon}}\} = \frac{prob(I[x])}{e^{\epsilon}}.$$

thus, $\epsilon_x \leqslant \epsilon$ holds for each $I[x]$. So, we prove that ϵ-transformed 0-DPkA can be achieved if ϵ-DPkA can be achieved.

Second, we are to prove that if ϵ-transformed 0-dp k-anonymity is achievable, then ϵ-dp k-anonymity is achievable.

Suppose ϵ-transformed 0-DPkA can be achieved by $0 \leqslant \epsilon_1, ..., \epsilon_{|I|} \leqslant \epsilon$, F_I^k and P_I^k. We are to prove that a randomized mechanism $\mathcal{M}(F_I^k, P_I^k)$ directly satisfies ϵ-DPkA. For each $F_I^k[i] \in F_I^k$ and any $I[j], I[j'] \in F_I^k[i]$, we have $\frac{prob(I[j]) \times prob(F_I^k[i]|I[j])}{e^{\epsilon_j}} = \frac{prob(I[j']) \times prob(F_I^k[i]|I[j'])}{e^{\epsilon_{j'}}}$ according to the definition of ϵ-transformed 0-dp k-anonymity. Then, we can get $\frac{prob(I[j]) \times prob(F_I^k[i]|I[j])}{prob(I[j']) \times prob(F_I^k[i]|I[j'])} = \frac{e^{\epsilon_j}}{e^{\epsilon_{j'}}} = e^{\epsilon_j - \epsilon_{j'}} \leqslant e^{\epsilon}$, since $0 \leqslant \epsilon_j, \epsilon_{j'} \leqslant \epsilon$. So, we see a randomized mechanism $\mathcal{M}(F_I^k, P_I^k)$ achieves ϵ-DPkA. Thus, we finish the second aspect of the proof. □

Theorem 4 demonstrates that we can solve a problem of achieving ϵ-transformed 0-DPkA with minimum ϵ to

obtain a solution for the problem of achieving the optimal DPkA. Next, we present a linear programming solution to compute $0 \leqslant \epsilon_1, ..., \epsilon_{|I|} \leqslant \epsilon$ for achieving ϵ-transformed 0-DPkA with minimum ϵ, given an instance $(I, prob, k, \epsilon)$. The sufficient and necessary condition for the availability of 0-DPkA is adopted in the formulation of linear constraints.

In the formulation of linear programming, we denote $x_i = \frac{1}{e^{\epsilon_j}}$ for each query interest $I[j] \in I$. Then, we have the following constraints for each $I[j] \in I$.

$$prob(I[j]) \times x_j - \frac{1}{k} \times \sum_{I[j'] \in I} \{prob(I[j']) \times x_{j'}\} \leqslant 0.$$

At the same time, denoting $x_0 = \frac{1}{e^{\epsilon}}$, then we have the following constraints for each $I[j] \in I$, since $0 \leqslant \epsilon_j \leqslant \epsilon$.

$$x_0 - x_j \leqslant 0$$
$$x_j - 1 \leqslant 0$$

Finally, due to the notation of x_0, we have the constraint that $x_0 - 1 \leqslant 0$. The subject is to get the minimum value of ϵ, and we calculate the minimized value of $-x_0$ to achieve this subject.

The formulation of our linear programming approach for computing $\epsilon_1, ..., \epsilon_{|I|}$ is as follow.

$$\begin{aligned} min \quad & -x_0 \\ s.t. \quad & prob(I[j]) \times x_j - \frac{1}{k} \times \\ & \sum_{I[j'] \in I} \{prob(I[j']) \times x_{j'}\} \leqslant 0 \qquad 1 \leqslant j \leqslant |I| \\ & x_0 - x_j \leqslant 0 \qquad 1 \leqslant j \leqslant |I| \\ & x_j - 1 \leqslant 0 \qquad 0 \leqslant j \leqslant |I| \\ & -x_j \leqslant 0 \qquad 0 \leqslant j \leqslant |I| \end{aligned}$$

There are $O(|I|)$ variables and $O(|I|)$ constraints in the linear programming we formulated above. After solving it, we get the value of $x_0, x_1, ..., x_{|I|}$ and we recover the value of ϵ as $-\ln x_0$. To compute F_I^k and P_I^k, we first transform the $prob(I[j])$ to $prob(I[j]) \times x_j$, and then invoke Algorithm 3 for solving 0-DPkA. According to the second part of proof for Theorem 4, we conclude that F_I^k and P_I^k returned by Algorithm 3 make up of a randomized mechanism which achieves ϵ-DPkA. Finally, we detail the process of achieving optimal DPkA in Algorithm 4.

Algorithm 4 *OptDP-k-Anonymity*

Input: query interest set I, integer $k < |I|$, prior probability $prob(I[i])$ for each $I[i] \in I$
Output: minimized ϵ, covering family of k-set F_I^k, probability assignment matrix P_I^k

```
x_0, x_1, ..., x_|I| = LinearProgSolver(I, prob(.), k);
prob'(.) = φ;
for j = 1, ..., |I| do
    prob'(.) = prob'(.) ∪ {(I[j], prob(I[j] × x_j))};
< F_I^k, P_I^k >= 0-DP-k-Anonymity(I, prob'(.));
ε = − ln x_0;
return ε, F_I^k, P_I^k;
```

Though the minimized ϵ is related to the prior probability of query interests in I, we should notice that the minimized ϵ returned in Algorithm 4 will not be larger than $\max_{j,j' \in I}\{\ln Prob(j) - \ln Prob(j')\}$ due to the result of Theorem 2. This provides a lower bound of privacy we can achieve in the worst case.

5 Simulation

We present the simulation settings and simulation results for our solution which achieves the optimal DPkA for given prior probability of each query interest in this section.

5.1 Simulation setup

We implement our approach to achieve the optimal DPkA in Java language, and we employ the function of solving linear programming from Apache Commons Mathematics Library [1]. We use four real-life datasets denoted TX, CA, $POIs$, and $Tweets$. TX and CA from [2] include street information in the state of Texas and California. Each entry in TX and CA contains latitude, longitude, and a set of keywords. $POIs$ and $Tweets$ from [10] contain worldwide coordinates and geo-tags. We take keywords and geo-tags as query interests in these datasets. To obtain the prior probability of each query interest, we first divide each dataset into regions with fixed sizes. We choose the size as $8km \times 8km$ and $30km \times 30km$ for both of TX and CA. Since $POIs$ and $Tweets$ cover worldwide area, we divide them with length $100km$ and $200km$. Given the size of I and a region R, I contains query interests with $|I|$-largest frequency in R. And the prior probability is calculated with in each region independently. In TX and CA , we remove entries only with keywords for city names and state names, since they dominate top frequency with few cardinality but provide no meaningful information. Details of datasets in our evaluation are listed in Table 2. To better understand

Table 2 Details of datasets in evaluation

Name	Description	Original/preprocessed size	Number of words
TX	Street objects in Texas	14182368/557918	64934
CA	Street objects in California	13820481/559349	70584
$POIs$	Pois worldwide	1157570/1157570	585626
$Tweets$	Tweets posted worldwide	20000000/20000000	2226543

the relation of the optimal ϵ and query interest distribution, we generate synthetic prior probability distribution of query interests following Zipf's law with exponent parameter sf as 0.01, 0.5, 1, 1.5, and 2.0. The synthetic distribution covers from uniform distribution to highly skewed distributions.

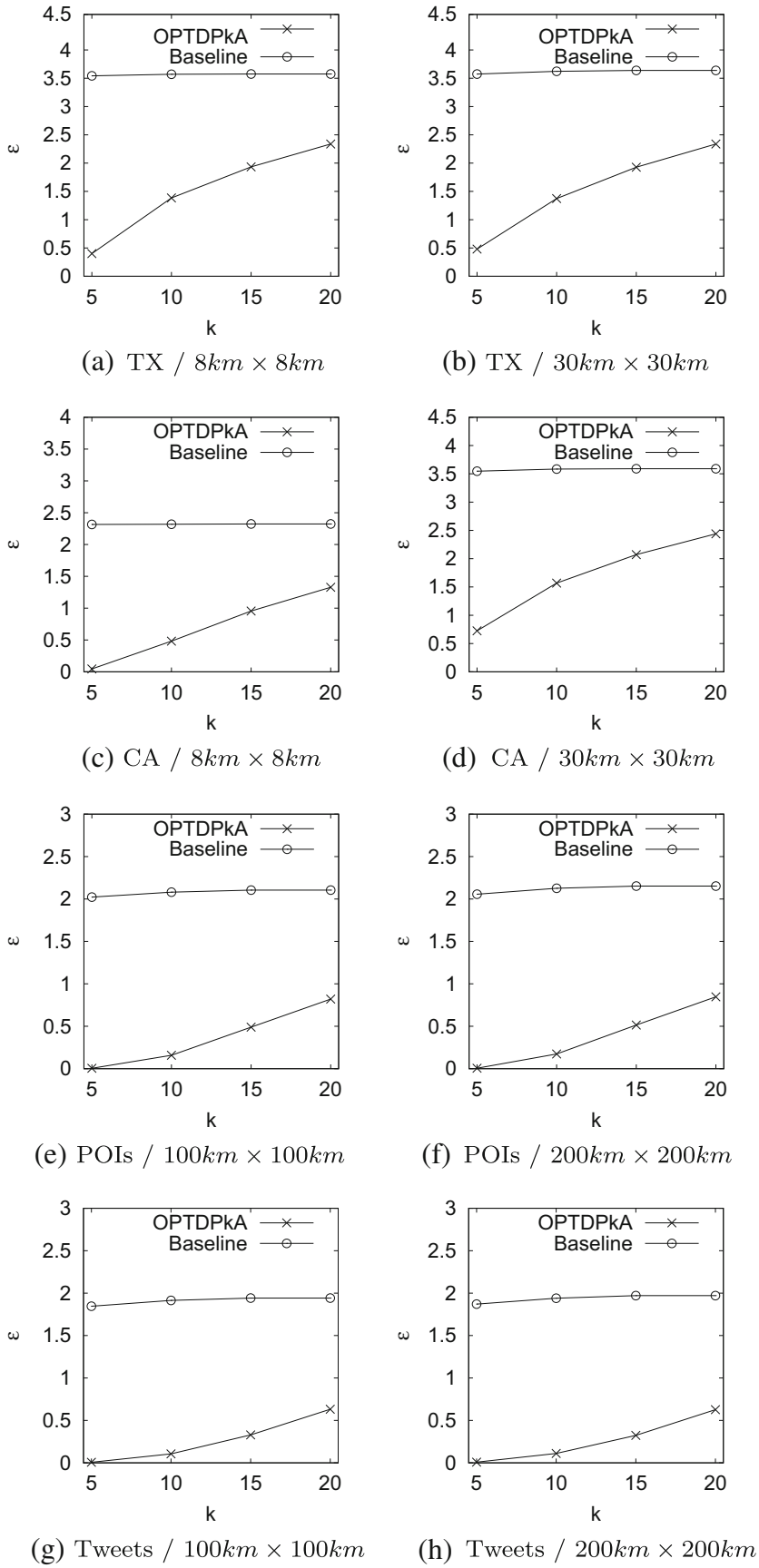
We evaluate the privacy budget ϵ achieved by our approach, which is minimized for a given prior probability of query interests. The mechanism presented in Algorithm 1 is compared to as the baseline, since it provides a lower bound of available privacy. We test effects of parameters on the minimized ϵ, including $|I|$ and k for the real-life datasets mentioned above. For $Zipf$ -distributed synthetic datasets, we include another parameter as the exponent value of Zipf distribution, which depicts the skewness of query interests. For a given $|I|$ and k, we traverse all the regions from the target datasets after division, and choose regions with more than $|I|$ distinct query interests for evaluation. Other regions with fewer than $|I|$ query interests are discarded for further tests. In each chosen region, we take the most frequent $|I|$ keywords or geo-tags as query interests and compute their probability among these $|I|$ query interests. In this way, we get the prior probability of each query interest in consideration. Then, the prior probability is feed to our proposed approach to obtain the mechanism for achieving the minimized ϵ.

5.2 Simulation result

In this section, we present the simulation results based on four real-life datasets and synthetic distributions. First, we study the effect of k on the available optimal privacy. To this end, we fix the parameter $|I| = 40$ and let k grow from 5 to 20, in other words, our setting includes 40 query interests and each query reports k of them. Figure 3 shows the effect of k on the optimal privacy in TX, CA, $POIs$, and $Tweets$. As illustrated, the minimized ϵ grows as k increases. The reason is that as more query interests are involved, the prior probability distribution becomes more skew, since newly included query interests appear much fewer times than top frequent ones. The growing skewness surely brings down the privacy, and the minimized ϵ grows. In all the tests shown in Fig. 3, our approach achieves better privacy than the baseline, especially when k is small. Our approach achieves ϵ close to 0 when k is small, meaning that very good privacy is obtained. As k grows to 20 which is half of total number of query interests, we still produce smaller ϵ by 2–3. Overall, the values of ϵ in $POIs$ and $Tweets$ are smaller than 1, even if k grows to 20. TX and CA provide small ϵ when k is 5, but they give ϵ larger than 1 for $k = 10, 15, 20$ in almost all the cases. For different region sizes, TX, $POIs$, and $Tweets$ produce close ϵ values for the same k, but CA provides larger ϵ for larger regions. This is due to the more skew nature of CA. For both small and large region tests, TX, $POIs$, and $Tweets$ provide similar number of regions for test, but CA shows large difference. Sixty regions are qualified when divided $8km \times 8km$, and 200 regions are qualified when divided $30km \times 30km$ for CA dataset. This means that the 140 newly included datasets for $30km \times 30km$ setting are more skew than the 60 regions for $8km \times 8km$ setting, thus they degrade ϵ.

Next, we test the effects of $|I|$ on ϵ. In this part, we test $|I|$ with different values including 30, 35, 40, and 45, and present the relation between $|I|$ and the minimized ϵ. At the same time, we study the effects of $|I|$ under three settings of k including $k = 5, k = 10$, and $k = 15$. In Fig. 4, we present the simulation results in TX, CA, $POIs$, and $Tweets$. From the results for each dataset with divided regions of different sizes, we conclude the trend that the minimized ϵ decreases as we increase the value of $|I|$. The reason is that increasing $|I|$ makes top-sized probability of query interests smaller. Due to the nature of our approach, such condition brings better privacy. At meanwhile, ϵ also degrades with k for a given value of $|I|$, and this is consistent with the former part of simulation as shown in Fig. 3. Again, TX, $POIs$, and $Tweets$ provide similar ϵ while CA provides larger ϵ for larger divided regions. The reason is the same as that of the trend presented in Fig. 3c, d. That is more skew probability of query interests are included when CA is divided into $30km \times 30km$ regions. The overall condition of ϵ in $POIs$ and $Tweets$ is better than that of TX and CA. Under different setting of $|I|$ and k, the values of ϵ for $POIs$ and $Tweets$ are no more than 0.6, which means good privacy is obtained. At the same time, when k is set to 10

Fig. 3 Effect of parameter k, $|I| = 40$



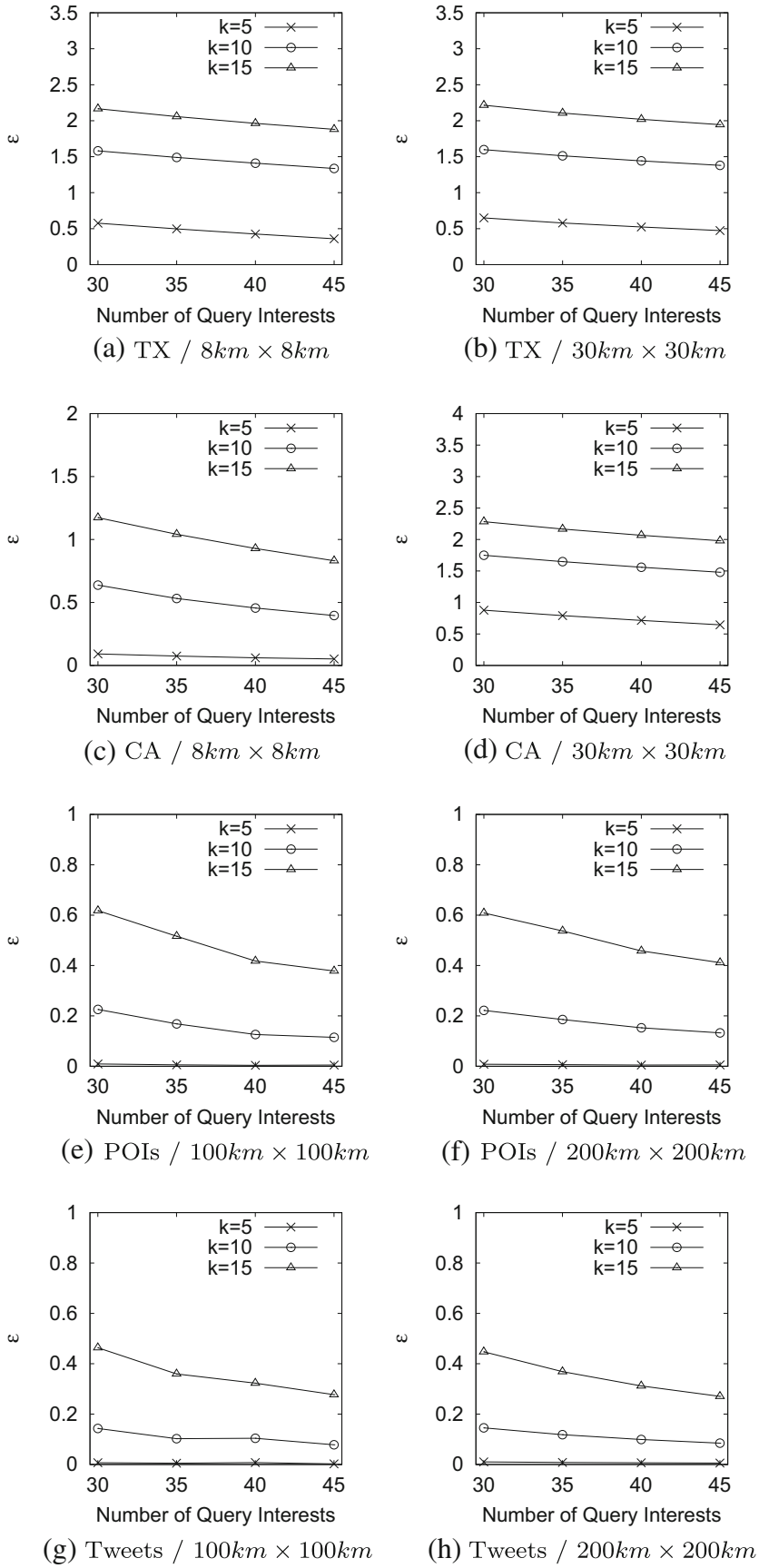
(a) TX / $8km \times 8km$ (b) TX / $30km \times 30km$

(c) CA / $8km \times 8km$ (d) CA / $30km \times 30km$

(e) POIs / $100km \times 100km$ (f) POIs / $200km \times 200km$

(g) Tweets / $100km \times 100km$ (h) Tweets / $200km \times 200km$

Fig. 4 Effect of parameter $|I|$



(a) TX / $8km \times 8km$ (b) TX / $30km \times 30km$

(c) CA / $8km \times 8km$ (d) CA / $30km \times 30km$

(e) POIs / $100km \times 100km$ (f) POIs / $200km \times 200km$

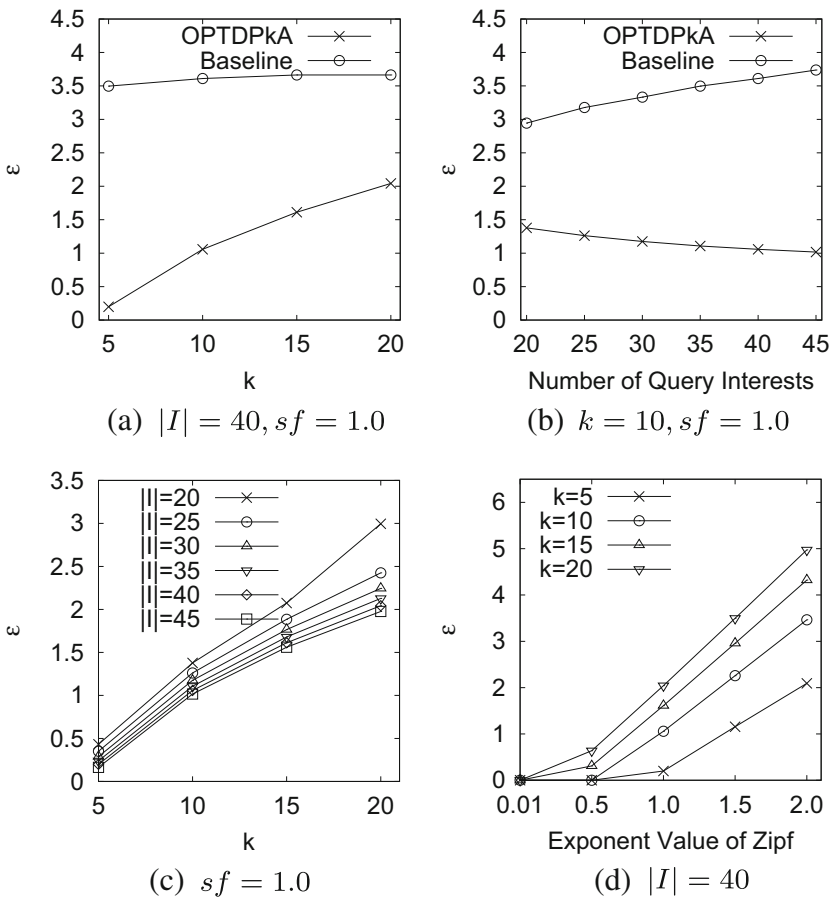
(g) Tweets / $100km \times 100km$ (h) Tweets / $200km \times 200km$

and 15, the values of ϵ are larger than 1 in TX (divided into $8km \times 8km$ and $30km \times 30km$ regions) and CA (divided into $30km \times 30km$). This is because $POIs$ and $Tweets$ are more uniform with regard to query interest probability. TX and CA are still more skew even we remove dominating entries contain only the city names and state names.

To further understand the effects of query interests' prior probability on ϵ, we generate $zipf$ distributed probability for query interests in I for different parameter $|I|$. During the generation, we set the exponent value of $zipf$ distribution (denoted sf) as 0.01 (almost uniform), 0.5, 1.0, 1.5, and 2.0 (highly skewed). When sf grows to 1.5 and 2.0, the generated distributions become very skew. For each distribution, we compute the ϵ achieved for different parameter k, so for each combination of $|I|$, sf , and k , we test the value of ϵ. Figure 5 shows the results of this part. In Fig. 5a, we fix $|I| = 40$ and $sf = 1.0$ and test different values of k. Consistent with former results shown in Fig. 3, ϵ grows with k and our approach again defeats the baseline. When k is small, ϵ is nearly 0. The value of ϵ for our approach is still smaller than that of the baseline by 3 when k grows to 20. Figure 5b shows the effect of $|I|$ on ϵ when we fix $sf = 1.0$ and $k = 10$. Again, we see ϵ decreases as $|I|$ increases. This is because increasing $|I|$ reduces top probabilities of query interests, which in turn brings opportunity for achieving smaller ϵ. Figure 5c shows more details on the effects of $|I|$ and k for our approach. ϵ grows with k, and it decreases as we increase $|I|$. Figure 5d studies the relation of ϵ and sf. As shown in Fig. 5d, ϵ stays on nearly 0 when the probability is almost uniform with $sf = 0.01$. When sf is set to 0.5, for different k values ϵ keeps smaller than 1. For other larger value of sf, ϵ goes beyond 1. In practice, the number of query interest could be of larger scale and k is of similar scale in our tests, so as demonstrated in our simulation proper ϵ will be achieved in reality.

6 Related work

Privacy has attracted significant attention from research community and a diversity of efforts has been devoted in this literature, including social network [9, 19, 42] and work on preserving privacy through smartphones, and they are horizontal to our work. Huang et al. [21] design protocols for preserving privacy for wireless sensor networks. Privacy in social network data is also widely studied [18].



(a) $|I| = 40, sf = 1.0$ (b) $k = 10, sf = 1.0$

(c) $sf = 1.0$ (d) $|I| = 40$

Fig. 5 Results on Zipf distributions

k-anonymity is first introduce in database community [34] and it is also well adopted for protecting privacy in LBS applications. A majority of works utilizes a trusted server for achieving k-anonymity through the technique of cloaking. Cloaking hides a user among other users inside a generated area, and LBS provider cannot recognize which user is submitting a query. This technique works for both $location\ privacy$ and $query\ privacy$. Works such as [5, 12, 36, 38] fall in this category. Niu et al. [28] work on the problem of generating proper dummies for locations in reported queries to LBS for hiding the user's actual locations. In [28], $2k$ locations with similar probability with the user's location are chosen as dummy candidates, and $k-1$ of them are randomly selected as final dummies. This approach obtains good entropy for the k locations in the reported query. Although this solution includes random nature, the posterior probability of the k reported location is still different due to the process of dummy selection, and the privacy guaranteed is not clear. Niu et al. [29] employ cache to avoid submitting queries to LBS server as much as possible, and thus prevent the leakage of user's location. Pingley et al. [30] propose a mechanism for protecting $query\ privacy$ in a continuous manner. A set of k query interests are generated for a traveling path, and the user submits the same queries along the path to avoid privacy breach. This fits to continuous querying; however, there is no privacy guarantee since it simply chooses query interests with probability larger than a given threshold as candidates. In summary, server-based k-anonymity suffers single point of failure and existing client-based solutions do not provide provable privacy guarantee based on the k-location/ query interests reported to LBS server. To our best knowledge, our work is the first to combine differential privacy and k-anonymity in the literature of LBS queries, and work on providing the optimal privacy guarantee that we can achieve. Zheng et al. [43] study the problem of protecting location privacy in local business service system. This work deals with privacy concerns in the process of recommendation such as [45].

Differential privacy is first introduced in statistic databases [13]. The intuitive idea of differential privacy is that a single change of the input should not modify the output significantly. By this guarantee, the adversary cannot recognize the input among all possible inputs similar to the real one. Due to the simple and clean nature of differential privacy, it has been adopted widely, such as machine learning [7, 33], statistic database [14, 16, 25], data mining [15, 41], graph [24], and crowdsourcing [35, 40]. Recent research starts combining correlation [27, 37, 39] and personality [23] nature to original differential privacy. Our work is parallel to the large body of differential privacy research. We combine differential privacy and k-anonymity to provide guaranteed privacy in LBS. Differential privacy has been adopted in the literature of privacy protection in LBS and [4] ensures that an adversary will not get significant information about a user's location after a query is reported. This is achieved by making the ratio of two nearby locations' posterior probability similar to that of their prior probability. Mechanisms following or adopting similar privacy guarantee are presented to optimize privacy or utility [6, 31]. Our approach differs from that of these works in several aspects. First, we work on protecting $query\ privacy$ in which similarity of different query interests does not make sense. Second, we combine differential privacy and k-anonymity to fit the scenario of protecting $query\ privacy$ in LBS, and optimize the privacy we can achieve. Last but not least, we provide privacy guarantee irrelevant with prior probability of query interests, as presented in Section 3. Thus, our notion leads to better privacy than that provided by a mirror variant of for $query\ privacy$. Other works employ customized measures for privacy in various aspects, such as [8, 20, 44] in social network datasets. Linear programming is adopted to compute the optimal privacy in our paper, and a recent study [3] works on evaluating the efficiency of a network structure using fractional programming. This work is horizontal to our paper.

7 Conclusion

In this paper, we propose a novel notion of differentially private k-anonymity, which is the first attempt to combine differential privacy and k-anonymity in LBS literature, for preserving $query\ privacy$ of LBS users. We recognize the lower bound of privacy available for DPkA by building a mirror variant of for $query\ privacy$. A sufficient and necessary condition under which 0-DPkA can be achieved is obtained in this paper. For more general cases that 0-DPkA is not achievable, we prove that the problem of achieving ϵ-DPkA is equivalent to the problem of achieving ϵ-transformed 0-DPkA defined in this paper. Then, we present an algorithm based on a linear programming technique to compute the optimal solution of the problem of achieving ϵ-transformed 0-DPkA, and in turn, we give the algorithm for achieving the optimal DPkA by minimizing ϵ. We conduct extensive simulation based on four real-life datasets and synthetic $zipf$ distributions, and the simulation results demonstrate the effectiveness of our approach for achieving well-defined notion of guaranteed $query\ privacy$.

To our best knowledge, this paper is the first work to combine differential privacy and k-anonymity to define and protect $query\ privacy$ in LBS scenarios. We provide guaranteed indistinguishability, which overcomes the major breach of traditional k-anonymity, among k query interests reported to LBS provider. Our work is a promising step

towards exploring practical privacy notion and preserving methods for *query privacy* in LBS applications.

Funding information This work is supported by Project (no. 61602129, 61632010, 61772157, U1509216) supported by the National Natural Science Foundation of China; This work is partly supported by the National Science Foundation (NSF) under grant NOs. 1252292, 1741277 and 1704287; China Postdoctoral Science Foundation Funded Project (grant no. 2014M561351); Heilongjiang Postdoctoral Science Foundation Funded Project (grant no. LBH-Z14118); and Sichuan Science and Technology Foundation-funded Project (grant no. 2017JZ0031).

References

1. Apache commons mathematics library, http://commons.apache.org/proper/commons-math/
2. Open street map, http://www.openstreetmap.org/
3. Ahmadzadeh R, Kordrostami S, Amirteimoori A (2017) Evaluating the efficiency of a two-stage network structure with the use of fractional programming. Discrete Mathematics. Algorithms Appl 09(03):1750,034. https://doi.org/10.1142/S1793830917500343
4. Andrés ME, Bordenabe NE, Chatzikokolakis K, Palamidessi C (2013) Geo-indistinguishability: differential privacy for location-based systems. In: Proceedings of the 2013 ACM SIGSAC conference on Computer & Communications Security, CCS '13. ACM, New York, pp 901–914
5. Bamba B, Liu L, Pesti P, Wang T (2008) Supporting anonymous location queries in mobile environments with privacygrid. In: Proceedings of the 17th international conference on world wide web, WWW '08. ACM, New York, pp 237–246
6. Bordenabe NE, Chatzikokolakis K, Palamidessi C (2014) Optimal geo-indistinguishable mechanisms for location privacy. In: Proceedings of the 2014 ACM SIGSAC conference on computer and communications security, CCS '14. ACM, New York, pp 251–262
7. Boyd K, Lantz E, Page D (2015) Differential privacy for classifier evaluation. In: Proceedings of the 8th ACM workshop on artificial intelligence and security, AISec '15. ACM, New York, pp 15–23
8. Cai Z, He Z, Guan X, Li Y (2017) Collective data-sanitization for preventing sensitive information inference attacks in social networks. IEEE Trans Dependable Secure Comput PP(99):1–1. https://doi.org/10.1109/TDSC.2016.2613521
9. Capurso N, Song T, Cheng W, Yu J, Cheng X (2017) An android-based mechanism for energy efficient localization depending on indoor/outdoor context. IEEE Internet Things J 4(2):299–307. https://doi.org/10.1109/JIOT.2016.2553100
10. Chen L, Cong G, Cao X, Tan KL (2015) Temporal spatial-keyword top-k publish/subscribe. In: 2015 IEEE 31St international conference on data engineering, pp 255–266. https://doi.org/10.1109/ICDE.2015.7113289
11. Chen X, Pang J (2013) Exploring dependency for query privacy protection in location-based services. In: Proceedings of the third ACM conference on data and application security and privacy, CODASPY '13. ACM, New York, pp 37–48. https://doi.org/10.1145/2435349.2435354
12. Chen X, Pang J (2014) Protecting query privacy in location-based services. GeoInformatica 18(1):95–133
13. Dwork C (2006) Differential privacy. In: 33Rd International Colloquium on Automata, Languages and Programming, part II (ICALP 2006). Springer, Venice
14. Dwork C (2008) Differential privacy: a survey of results. Springer, Berlin
15. Friedman A, Schuster A (2010) Data mining with differential privacy. In: Proceedings of the 16th ACM SIGKDD international conference on knowledge discovery and data mining, KDD '10. ACM, New York, pp 493–502
16. Haney S, Machanavajjhala A, Ding B (2015) Design of policy-aware differentially private algorithms. Proc VLDB Endow 9(4):264–275
17. He Z, Cai Z, Sun Y, Li Y, Cheng X (2017) Customized privacy preserving for inherent data and latent data. Personal Ubiquitous Comput 21(1):43–54. https://doi.org/10.1007/s00779-016-0972-2
18. He Z, Cai Z, Wang X (2015) Modeling propagation dynamics and developing optimized countermeasures for rumor spreading in online social networks. In: 2015 IEEE 35Th international conference on distributed computing systems, pp 205–214. https://doi.org/10.1109/ICDCS.2015.29
19. He Z, Cai Z, Yu J (2017) Latent-data privacy preserving with customized data utility for social network data. IEEE Trans Vehicular Technol PP(99):1–1. https://doi.org/10.1109/TVT.2017.2738018
20. He Z, Cai Z, Yu J, Wang X, Sun Y, Li Y (2017) Cost-efficient strategies for restraining rumor spreading in mobile social networks. IEEE Trans Veh Technol 66(3):2789–2800. https://doi.org/10.1109/TVT.2016.2585591
21. Huang H, Gong T, Chen P, Malekian R, Chen T (2016) Secure two-party distance computation protocol based on privacy homomorphism and scalar product in wireless sensor networks. Tsinghua Sci Technol 21(4):385–396. https://doi.org/10.1109/TST.2016.7536716
22. Jha SK (2017) Revisiting calculation of moments of number of comparisons used by the randomized quick sort algorithm. Discrete Mathematics. Algorithms Appl 09(01):1750,001. https://doi.org/10.1142/S179383091750001X
23. Jorgensen Z, Yu T, Cormode G (2015) Conservative or liberal? Personalized differential privacy. In: 2015 IEEE 31St international conference on data engineering, pp 1023–1034
24. Kasiviswanathan SP, Nissim K, Raskhodnikova S, Smith A (2013) Analyzing graphs with node differential privacy. In: Proceedings of the 10th theory of cryptography conference on theory of cryptography, TCC'13. Springer, Berlin, pp 457–476
25. Kifer D, Machanavajjhala A (2011) No free lunch in data privacy. In: Proceedings of the 2011 ACM SIGMOD international conference on management of data, SIGMOD '11. ACM, New York, pp 193–204
26. Liang Y, Cai Z, Han Q, Li Y (2017) Location privacy leakage through sensory data. Security and Communication Networks
27. Liu C, Chakraborty S, Mittal P (2016) Dependence makes you vulnerable: differential privacy under dependent tuples. In: Proceedings of the network and distributed system security symposium 2016 (NDSS), pp 0-0, San Diego, California, USA
28. Niu B, Li Q, Zhu X, Cao G, Li H (2014) Achieving k-anonymity in privacy-aware location-based services. In: INFOCOM
29. Niu B, Li Q, Zhu X, Cao G, Li H (2015) Enhancing privacy through caching in location-based services. In: INFOCOM
30. Pingley A, Zhang N, Fu X, Choi HA, Subramaniam S, Zhao W (2011) Protection of query privacy for continuous location based services. In: 2011 Proceedings of IEEE INFOCOM, pp 1710–1718
31. Shokri R, Theodorakopoulos G, Troncoso C, Hubaux JP, Le Boudec JY (2012) Protecting location privacy: optimal strategy against localization attacks. In: Proceedings of the 2012 ACM conference on computer and communications security, CCS '12. ACM, New York, pp 617–627
32. Song T, Capurso N, Cheng X, Yu J, Chen B, Zhao W (2017) Enhancing GP with lane-level navigation to facilitate highway driving. IEEE Trans Veh Technol 66(6):4579–4591. https://doi.org/10.1109/TVT.2017.2661316
33. Stoddard B, Chen Y, Machanavajjhala A (2014) Differentially private algorithms for empirical machine learning. arXiv:1411.5428

34. Sweeney L (2002) K-anonymity: a model for protecting privacy. Int J Uncertain Fuzziness Knowl.-Based Syst 10(5):557–570
35. To H, Ghinita G, Shahabi C (2014) A framework for protecting worker location privacy in spatial crowdsourcing. Proc VLDB Endow 7(10):919–930
36. Wang Y, Xu D, Li F (2016) Providing location-aware location privacy protection for mobile location-based services. Tsinghua Sci Technol 21(3):243–259. https://doi.org/10.1109/TST.2016.7488736
37. Xiao Y, Xiong L (2015) Protecting locations with differential privacy under temporal correlations. In: Proceedings of the 22nd ACM SIGSAC conference on computer and communications security (CCS), pp 1298–1309, enver, Colorado, USA
38. Xue M, Kalnis P, Pung HK (2009) Location diversity: enhanced privacy protection in location based services. In: Proceedings of the 4th international symposium on location and context awareness, loCA '09. Springer, Berlin, pp 70–87
39. Yang B, Sato I, Nakagawa H (2015) Bayesian differential privacy on correlated data. In: Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data (SIGMOD), pp 747–762, Melbourne, Victoria, Australia
40. Wang Y, Cai Z, Ying G, Gao Y, Tong X, Wu G (2016) An incentive mechanism with privacy protection in mobile crowdsourcing systems. Comput Netw 102(Supplement C):157–171. https://doi.org/10.1016/j.comnet.2016.03.016
41. Zeng C, Naughton JF, Cai JY (2012) On differentially private frequent itemset mining. Proc VLDB Endow 6(1):25–36
42. Zhang L, Cai Z, Wang X (2016) Fakemask: a novel privacy preserving approach for smartphones. IEEE Trans Netw Serv Manag 13(2):335–348. https://doi.org/10.1109/TNSM.2016.2559448
43. Zheng X, Cai Z, Li J, Gao H (2017) Location-privacy-aware review publication mechanism for local business service systems. In: 2017 Proceedings of IEEE INFOCOM
44. Zheng X, Cai Z, Yu J, Wang C, Li Y (2017) Follow but no track: privacy preserved profile publishing in cyber-physical social systems. IEEE Internet Things J PP(99):1–1. https://doi.org/10.1109/JIOT.2017.2679483
45. Zhou Z, Cheng Z, Zhang LJ, Gaaloul W, Ning K (2017) Scientific workflow clustering and recommendation leveraging layer hierarchical analysis. IEEE Trans Services Comput PP(99):1–1. https://doi.org/10.1109/TSC.2016.2542805