计算机研究与发展 ISSN 1000-1239/CN 11-1777/TP
Journal of Computer Research and Development 47(1): 121-129, 2010

基于位置服务中的连续查询隐私保护研究

潘 晓 郝 兴 孟小峰
(中国人民大学信息学院 北京 100872)
(smallpx@ruc.edu.cn)

Privacy Preserving Towards Continuous Query in Location-Based Services

Pan Xiao, Hao Xing, and Meng Xiaofeng
(School of Information, Renmin University of China, Beijing 100872)

Abstract With advances in wireless communication and mobile positioning technologies, location-based mobile services have been gaining increasingly popularity in recent years. Privacy preservation, including location privacy and query privacy, has recently received considerable attention for location-based mobile services. A lot of location cloaking approaches have been proposed for protecting the location privacy of mobile users. However, they mostly focus on anonymizing snapshot queries based on proximity of locations at query issued time. Therefore, most of them are ill-suited for continuous queries. In view of the privacy disclosure (including location and query privacy) and poor quality of service under continuous query anonymization, a δ_t-privacy model and a δ_t-distortion model are proposed to balance the tradeoff between privacy preserving and quality of service. Meanwhile a temporal distortion model is proposed to measure the location information loss during a time interval, and it is mapped to a temporal similar distance between two queries. Finally, a greedy cloaking algorithm (GCA) is proposed, which is applicable to both anonymizing snapshot queries and continuous queries. Average cloaking success rate, cloaking time, processing time and anonymization cost for successful requests are evaluated with increasing privacy level (k). Experimental results validate the efficiency and effectiveness of the proposed algorithm.

Key words privacy; continuous query; quality of service; LBS; mobile computing

摘 要 近年来,伴随着移动计算技术和无限设备的蓬勃发展,位置服务中的隐私保护研究受到了学术界的广泛关注,提出了很多匿名算法以保护移动用户的隐私信息.但是现有方法均针对 snapshot 查询,不能适用于连续查询.如果将现有的静态匿名算法直接应用于连续查询,将会产生隐私泄露、匿名服务器工作代价大等问题.针对这些问题,提出了 δ_t-隐私模型和 δ_t-质量模型来均衡隐私保护与服务质量的矛盾,并基于此提出了一种贪心匿名算法.该算法不仅适用于 snapshot 查询,也适用于连续查询.实验结果证明了算法的有效性.

关键词 隐私;连续查询;服务质量;基于位置服务;移动计算

中图法分类号 TP392

收稿日期: 2009-06-26; 修回日期: 2009-09-29

基金项目: 国家自然科学基金项目(60833005, 60573091);国家"八六三"高技术研究发展计划基金项目(2007AA01Z155, 2009AA011904);高等学校博士学科点专项科研基金项目(20080020002)

随着移动计算技术和无线设备的结合, 随时随地获得个人精确位置成为可能, 促进了新一类应用程序——位置服务(location based service, LBS)的产生和发展. 但是, 人们在享受各种位置服务带来便捷的同时, 个人隐私信息泄漏问题逐渐引起广大学者的关注, 成为近年来研究的热点问题之一[1].

一般来讲, 位置服务中的隐私保护可以分为两种: 位置隐私[2]和查询隐私[3]. 如张某利用自己带有GPS的手机提出"寻找5 min内距离我最近的肿瘤医院". 这是导航系统中常见的连续最近邻查询. 一方面, 用户不想让任何人知道他现在所在位置(如"医院"); 另一方面用户也不想让任何人获知自己提出了哪方面的查询请求, 如与某特定肿瘤相关的医院查询. 前者属于位置隐私保护范畴, 后者属于查询隐私保护范畴.

为了解决位置服务中的隐私保护问题, Gruteser等人[4]提出了位置k-匿名模型: 当一个移动用户的位置无法与其他$(k-1)$个用户的位置相区别时, 称此位置满足位置k-匿名. 此模型既适用于位置隐私保护, 也适用于查询隐私保护. 以位置隐私为例, 如图1(a)所示, 用户A, B, C匿名后的位置用矩形R表示, 攻击者仅知道R中包含3个用户, 但无法确定每个用户的确切位置, 从而保护了用户的位置隐私. 类似地, 用户提出的查询隐私可以通过相同的方法得到保护. 为了解决查询隐私保护中查询差异性问题, Xiao Zhen等人[5]提出了p-敏感模型. 此模型考虑了查询敏感度和语义差异性, 要求在一个匿名集中敏感查询个数所占比例不能超过p. 如图1(b)所示, 矩形R中提出了3种查询: {肿瘤医院, 旅馆, 加油站}, 其中有关肿瘤医院的查询具有敏感性, 用户$A/B/C$提出此查询的概率均为1/3. 如果$p=1/2$, 则在这个例子中满足1/2-敏感模型.

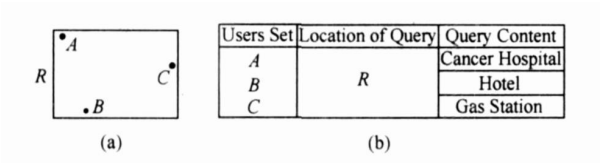

Fig. 1 Privacy in LBS. (a) Location privacy and (b) Query privacy.

图1 位置服务中的隐私保护. (a) 位置隐私; (b) 查询隐私

位置服务中现有的隐私保护工作均针对snapshot查询类型. 然而, 连续查询是位置服务中一种常见并重要的查询类型, 具有位置频繁更新和时效性①的特点. 如果将现有的匿名算法直接应用于连续查询隐私保护将产生以下3个问题: 第一, 连续查询隐私泄露, 如图2所示, 系统中存在{A, B, C, D, E, F} 6个用户, 攻击者知道存在某个连续查询, 但并不知道连续查询是什么以及是由谁提出的, 在3个不同时刻t_i, t_{i+1}, t_{i+2}, 用户A形成了3个不同的匿名集, 即{A, B, D}, {A, B, F}, {A, C, E}, 如图2中实线矩形框所示, 将3个匿名集取交, 即可获知是用户A提出的连续查询以及此连续查询类型; 第二, 加剧了匿名服务器的负担. 移动对象连续发生位置更新, 并且每发生一次更新均需要为新位置重新生成一个匿名框, 造成了匿名服务器负担过重, 从而变成系统瓶颈; 第三, 很多网络资源被浪费于传输频繁的位置更新和新生成的匿名集, 造成网络拥堵.

上述问题主要是由同一用户(A)在其有效生命期内形成的匿名集不同而造成的. 所以最简单方法是让连续查询的用户在最初时刻形成的匿名集在其查询有效期内均有效. 在前面的例子中, 用户A在t_i时刻形成的匿名集是{A, B, D}, 则在t_{i+1}, t_{i+2}时刻匿名集依然是{A, B, D}, 如图2中虚线矩形所示. 很明显, 这种方法将产生新的问题是: 第一, 位置隐私泄露, 如在图2(b)中, 在t_{i+1}时刻, A, B, D位置过于邻近, 造成匿名框过小(极端情况下集中于一点), 位置隐私泄露; 第二, 服务质量QoS(quality of services)降低. 服务质量与数据精度成反比. {A, B, D}在t_{i+2}时刻分布在距离较远的位置, 形成的匿名框过大, 造成过高的查询处理代价.

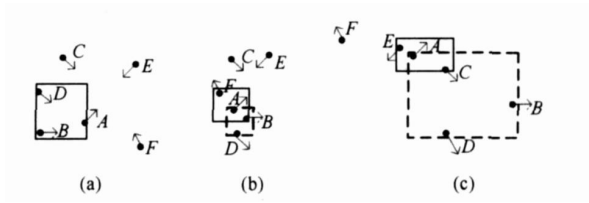

Fig. 2 Privacy preserving for continuous query. (a) Cloaking set at t_i; (b) Cloaking set at t_{i+1}; and (c) Cloaking set at t_{i+2}.

图2 连续查询隐私保护. (a) 在时刻t_i的匿名集; (b) 在时刻t_{i+1}的匿名集; (c) 在时刻t_{i+2}的匿名集

由此可见, 仅仅简单地把在最初时刻形成的匿名集作为连续查询有效期内的匿名集返回并不能解决问题. 其主要原因是现有的匿名算法仅考虑用户当前位置的邻近性, 忽略了用户未来的位置. 用户

① 时效性是指查询具有一定的生命周期, 在特殊情况下, snapshot查询其生命周期为0.

位置邻近性会随着移动用户的运动而改变. 更具体地说, 当前邻近的对象可能在下一个时刻随着对象的散开而距离很远; 而当前距离很远的对象可能在下一个时刻重合于一个点. 所以, 此问题的难点在于: 在用户查询的有效期内选取哪一个时刻的位置进行匿名, 形成的匿名集既不会产生位置隐私泄露, 也不会产生查询隐私泄露①, 同时可以保证最好的服务质量.

针对这一问题, 提出了 δ_t-隐私模型和 δ_q-质量模型来均衡(trade-off)隐私保护与服务质量这一矛盾; 通过匿名框的周长形式化定义匿名位置的可用性, 并将其定义为两个移动对象间的时序相似性, 利用两个对象的相似性提出了一个贪心算法, 从而保护用户的位置隐私和查询隐私. 该匿名算法既适用于 snapshot 查询, 也适用于连续查询.

1　相关工作

位置隐私保护和查询隐私保护是移动计算环境中主要考虑的两个隐私问题. 最初人们并没有把位置隐私与查询隐私分开, 而是将其合二为一地看待, 即保护了位置隐私等同于保护了查询隐私. 位置 k-匿名模型是学术界广泛接受的匿名模型, 由 Gruteser 等人提出, 随后很多人对此模型进行了修正[5-6].

位置匿名的基本思想分为 3 种: 第一, 发布假位置 (dummy)[7], 即不发布真实服务请求的位置, 假位置和真实位置的距离与隐私保护程度成正比, 和服务质量成反比; 第二, 时空匿名(spatial-temporal cloaking)[2,4], 本质上是降低移动对象位置的时空粒度, 即用时空区域表示用户真实的精确位置, 区域形状不限, 可以是任意形状的凸多边形, 称此匿名区域为匿名框(cloaking region), 匿名框的大小与匿名程度成正比, 与服务质量成反比; 第三, 加密(encryption)[8], 查询点与查询结果对服务提供商来说都是隐秘的. 这里采用的方法属于时空匿名.

文献[3]首次提出了连续查询隐私保护问题, 指出将用户在初始时刻形成的匿名集作为查询有效期内的最终结果, 从而解决连续查询隐私泄露的问题. 但是该工作存在以下缺点: 第一, 如引言部分所述, 该算法生成的匿名集仅考虑移动对象初始时刻位置的邻近性, 忽略对象的运动, 造成位置隐私泄露和糟糕的服务质量; 第二, 该算法要求任何新提出的查询一定要延迟一段时间才能匿名, 从而保证任何一个查询都是从已在系统中存在一定时间的旧用户提出的, 这样的做法造成用户的服务无法即时获得响应, 即使附近已有足够多的用户供其形成匿名集; 第三, 匿名集的形成忽略各个查询有效期的时间差异性, 造成同一匿名集中的查询有效期相差过大. 最坏情况下, 连续查询与 snapshot 查询匿名在一起, 只要有一个用户查询终止, 在此匿名集中的所有查询也被迫终止. 与该工作相比, 提出的算法与其不同点在于: 第一, 考虑了查询生命周期内每一个时刻位置的邻近性; 第二, 被匿名在一起的查询具有时效性相似的特点, 即查询有效期类似; 第三, 同时适用于连续查询和 snapshot 的查询. 文献[9]也解决了连续查询隐私保护的问题. 它假设在匿名区域内, 用户位置并非均匀分布, 采用信息理论中的熵(entropy)来定义用户的隐私保护度. 但是由于熵并不考虑用户的位置是否不同, 可能造成 k 个用户重叠于一点的情况, 从而产生位置隐私泄露.

2　系统结构

与大部分现有工作[2,4]一样, 采用中心服务器结构如图 3 所示, 包括移动用户、匿名服务器和服务提供商. 处理流程为: 移动用户将查询请求 Q 发送给匿名服务器. 查询请求分为新查询(new query)和活动查询(active query)两种. 新查询是指由用户首次提出的查询请求; 活动查询是指用户在过去的时刻提出、现在依然有效的查询, 再次触发仅为位置更新.

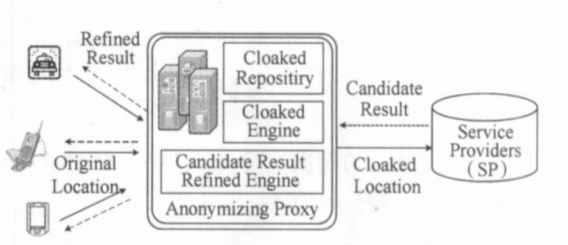

Fig. 3　System architecture for location privacy preserving.

图 3　位置隐私保护系统结构

1) 匿名服务器由知识库(cloaked repository)、匿名引擎(cloaked engine)和查询结果求精处理器(candidate result refined engine)组成. 对于新查询, 匿名引擎触发匿名算法根据系统待匿名查询的当前位置寻找匿名集, 并将匿名集发送到知识库和服务

① 匿名集中查询差异性并不属于本文解决范畴. 此问题可以用现有的 p-敏感模型加以解决.

提供商; 如果是活动查询, 则直接从知识库中寻找该查询在提出之初形成的匿名集合. 找到该集合, 并根据其中所有对象的当前位置计算新的匿名框, 发送给服务提供商.

2) 服务提供商接收到用户匿名后的位置进行查询处理, 并将查询结果发送给匿名服务器.

3) 匿名服务器中的查询结果求精处理器将对查询结果求精后返回给移动用户.

3 预备知识

定义 **1**. 查询 Q. 形式化地表示每一个查询 Q 为

$$Q = (l, \boldsymbol{v}, t, T_{\exp}, con),$$

其中:

① $l = (x, y)$ 表示查询 Q 所在位置;

②速度 $\boldsymbol{v} = (v_x, v_y)$ 是一个向量, 其中 v_x/v_y 表示查询在 x/y 轴方向上的运动速度分量;

③ $(l, \boldsymbol{v}, t)$ 表示查询 Q 在时刻 t 的位置在 l 上, 并且运动速度为 $\boldsymbol{v}$;

④ $T_{\exp}$ 表示该查询过期时间;

⑤ con 表示查询内容, 如最近医院等.

定义 **2.** 匿名集. 形式化的定义匿名集 CR:

$$(CID, Qset, R_{L,t}, R_{v,t}),$$

其中:

① CID 表示匿名集的标识符;

② $Qset$ 是一个集合, 由匿名集中包含的查询组成;

③ $R_{L,t} = (L_{x-,t}, L_{y-,t}, L_{x+,t}, L_{y+,t})$ 表示匿名框, 是覆盖 $Qset$ 中所有用户的最小边界矩形(minimum boundary rectangle, MBR), 其中, $(L_{x-,t}, L_{y-,t})$ 和 $(L_{x+,t}, L_{y+,t})$ 是 MBR 的左下角和右上角在时刻 t 的坐标.

④ $R_{v,t}$ 是 $R_{L,t}$ 的速度边界矩形(boundary velocity rectangle, BVR). $R_{v,t} = (v_{x\min,t}, v_{y\min,t}, v_{x\max,t}, v_{y\max,t})$, 其中 $v_{x\min,t} = \min(v_{x+,t}, v_{x-,t})$, $v_{x\max,t} = \max(v_{x+,t}, v_{x-,t})$, $v_{y\min,t} = \min(v_{y+,t}, v_{y-,t})$, $v_{y\max,t} = \max(v_{y+,t}, v_{y-,t})$. $v_{x-,t}/v_{x+,t}$ 是 MBR 在 x 方向上的左/右边界速度, $v_{y-,t}/v_{y+,t}$ 是 MBR 在 y 方向上的下/上边界速度.

注意 $v_{x\max,t}/v_{y\max,t}$ 和 $v_{x\min,t}/v_{y\min,t}$ 不一定是 $QSet$ 中 x/y 方向上的最小与最大速度. 如图 4 所示匿名集包括 $Q_1 \sim Q_5$ 五个查询, 括号中的数字表示该查询的运动速度. 很明显, 此时 x 轴方向上的 $v_{x\max} = 1$, 但并不是在该方向上的最大速度 $v_{\max} = 3$. 但是随着查询的运动, 存在时刻 t_j, Q_5 超越 Q_3, 此时边界速度 $v_{x\max,tj} = 3$. 所以边界速度会随着查询的运动而改变, MBR 边界的运动是一个分段函数, 如图 5 所示.

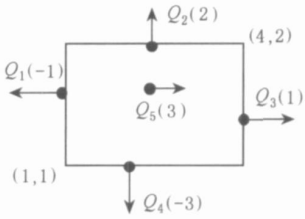

Fig. 4 Boundary velocities.

图 4 边界速度图

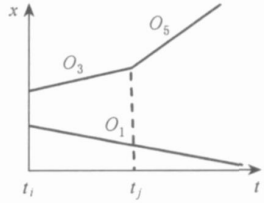

Fig. 5 WB on x-axis.

图 5 x 轴上的边界宽(WB)

定义 **3.** 边界长与宽. 假设匿名集 CR 在时刻 t 的匿名框为 $R_{L,t}$, 则在 x 轴上匿名框的宽为

$$WB_t = L_{x+,t} - L_{x-,t} = (L_{x+,t_i-1} - L_{x-,t_i-1}) + (v_{x+,t} - v_{x-,t}) \times (t - t_{i-1}). \quad (1)$$

同样地, 在 y 轴上匿名框的高为

$$HB_t = L_{y+,t} - L_{y-,t} = (L_{y+,t_i-1} - L_{y-,t_i-1}) + (v_{y+,t} - v_{y-,t}) \times (t - t_{i-1}), \quad (2)$$

分别记为 WB_t, HB_t.

采用降低位置信息空间粒度的方法保护位置隐私, 显然, 位置信息的粒度越低隐私保护度则越高, 但是数据可用性就越差. 所以我们使用信息扭曲度(distortion)来评价位置数据可用性. 数据扭曲度越高数据可用性则越差.

定义 **4.** 位置扭曲度(distortion). 假设匿名集 CR, 其匿名框是 $R_{L,t}$. 查询 $Q \in CR$, 在时刻 t, 查询 Q 的位置 l 被概化为 $R_{L,t} = (L_{x-,t}, L_{y-,t}, L_{x+,t}, L_{y+,t})$, 则位置扭曲度定义为

$$Distortion_{R_{v,t}}(Q, R_{L,t}) = \frac{(L_{x+,t} - L_{x-,t}) + (L_{y+,t} - L_{y-,t})}{A_{\text{height}} + A_{\text{width}}},$$

其中 A_{height}, A_{width} 是整个空间的高与宽, $(L_{x-,t}, L_{y-,t})$ 和 $(L_{x+,t}, L_{y+,t})$ 是匿名框 $R_{L,t}$ 在时刻 t 的左下和右上角坐标. 查询 Q 的有效期截至到时刻 $T_{\exp}$, 则 Q 在其有效期内总信息扭曲度为

$$\int_{T_s}^{T_{\exp}} Distortion_{R_{v,t}}(Q, R_{L,t})\,dt,$$

其中 T_s 是查询 Q 匿名成功的时刻.

定义 **5.** 匿名集的位置扭曲度. 匿名集 CR 的匿名框的边界位置和速度分别为 $R_{L,t}=(L_{x-,t}, L_{y-,t}, L_{x+,t}, L_{y+,t})$, $R_{v,t}=(v_{x\min,t}, v_{y\min,t}, v_{x\max,t}, v_{y\max,t})$. CR 在时刻 t 的位置扭曲度为 CR 中所有查询的扭曲度加和:

$$Distortion_{R_{v,t}}(CR, R_{L,t}) = \sum_{i=1}^{|CR|} Distortion_{R_{v,t}}(Q_i, R_{L,t}) = |CR| \frac{(L_{x+,t}-L_{x-,t})+(L_{y+,t}-L_{y-,t})}{A_{\text{height}}+A_{\text{width}}},$$

其中 $Q_i \in CR$. 在 CR 的有效期内总信息扭曲率为

$$\int_{T_s}^{maxT} Distortion_{R_{v,t}}(CR, R_{L,t})\,dt,$$

其中 T_s 是匿名集 CR 的生成时间, $maxT=\max_{Q\in CR} Q.T_{\exp}$.

很明显, 移动对象的状态(包括初始位置和速度)越相似, 其匿名在一起信息扭曲率则越低; 状态差异越大, 匿名后的信息扭曲率则越高. 所以扭曲率定义任意两个查询在其生命周期内的相似度.

定义 **6.** 时序相似度. Q_1 和 Q_2 是两个查询, $R_{L\_12,t}$ 是时刻 t 覆盖这两个查询的 MBR, 则 Q_1 与 Q_2 在其生命有效期内的相似度为

$$SimDis(Q_1, Q_2) = \int_{T_s}^{maxT} Distortion_{R_{v\_12,t}}(CS, R_{L\_12,t})\,dt,$$

其中 $maxT=\max(Q_1.T_{\exp}, Q_2.T_{\exp})$, 并且 $R_{v\_12,t}$ 是 MBR 在时刻 t 的边界速度集.

4　隐私模型

本节首先讨论一维的情况: 将查询 Q 的位置和速度分别向 x, y 轴投影. 例如图 6(a)所示, 一个匿名集包含{Q_1, Q_2, Q_3}3 个查询, 直线斜率表示查询在 x 轴方向上的速度. 从图中可以看出, 在 t_w 时刻, 3 个查询具有相同的 x 坐标, 该匿名集的一维信息泄露. 同理, y 轴也存在类似的情况. 最坏情况下, 匿名框从 x, y 方向上同时收缩, 并缩为一点, 如图 6(b)所示, 此时查询位置隐私泄露. 但是, 无论哪一种情况我们都认为是不允许的. 精确位置泄露可以看作是位置一维信息泄露的特殊情况. 所以, 只要保证匿名框的长和宽在查询有效期内的任何一个时刻均不会缩小为一点则可以保证位置隐私不泄露.

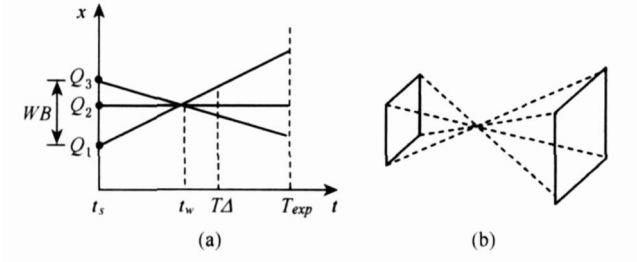

Fig. 6　Location privacy disclosed for continuous query. (a) In one dimension and (b) In two dimension.

图 6　连续查询位置隐私泄露示例. (a) 一维情况; (2) 二维情况

定义 **7.** δ_p-隐私模型. 设 WB_t/HB_t 是匿名框在时刻 t 的宽/高, δ_p 是用户指定的一维情况下最小的位置粒度, 则

$$\forall t \in [T_s, maxT], \min(WB_t, HB_t) \geqslant \delta_p \times P_A,$$

其中 $P_A = A_{\text{width}} + A_{\text{height}}$, 称该匿名集满足 δ_p-隐私模型.

匿名集除要满足最低隐私需求 δ_p 之外, 其位置信息扭曲度也不能过高, 否则影响服务质量. 所以, 为保证服务质量, 用户定义最高信息扭曲度 δ_q. 由于移动对象的运动, 位置扭曲度随时间不断变化. 匿名集 CR 在时刻 t_i 的信息丢失率不高于 δ_q并不代表在查询有效期内一直大于 δ_t.

定义 **8.** δ_r-质量模型. 假设用户可以容忍的最差服务质量是 δ_t, 匿名集 CR 的位置匿名框为 $R_{L,t}$, 伴随边界速度 $R_{v,t}$, 则对于

$$\forall t \in [T_s, maxT], \forall Q \in CR, Distortion_{R_{v,t}}(Q, R_{L,t}) \leqslant \delta_t,$$

则称该匿名集满足 δ_t-质量模型.

为简便起见, 假设系统具有统一的隐私度需求 k, 综上所述, 成功的匿名集需要满足以下 3 个条件:

1) $|CR| \geqslant k$;

2) 设 $minT=\min_{Q\in CR} Q.T_{\exp}$, $maxT=\max_{Q\in CR} Q.T_{\exp}$, $maxT-minT<\delta_t$;

3) 匿名集 CR 满足 δ_p-隐私模型和 δ_r-质量模型.

其中, 第 1 个条件符合位置 k-匿名模型, 要求在一个匿名集中至少包含 k 个查询; 第 2 个条件保证了同在一个匿名集中的查询, 具有时效相似的特点, 即查询有效期的差距不大于 δ; 第 3 个条件试图在隐私保护和服务质量上寻找平衡点.

5　贪心匿名算法

算法的主要思想是当新查询 r(以后称此查询为触发查询)到来时, 依次扫描待查询集合 $RSet$ 中

待匿名且并未过期的查询. 判断如果这两个查询形成匿名集, 是否满足 δ_t-质量模型. 如果满足则计算这两个查询时序相似度, 将 r 与具有最小相似度的查询聚集在一起; 如果不满足则取下一个查询. 重复上述过程, 直至包含该查询的候选匿名集从 $RSet$ 再也找不到合并的查询. 最后, 如果候选匿名集的大小, 即包含的用户数大于隐私度 k, 则调用算法 3, 判断该候选集是否满足 δ_p-隐私模型. 具体算法参见算法 1.

在算法 1 中包含 3 个重要步骤: 候选匿名集边界对象检测、δ_t-质量模型检测和 δ_p-隐私模型检测. 后面 3 节将逐一介绍.

算法 1. 贪心匿名算法(GCA).

/* 当有新查询 r 到来时* /
① candidate cloaking set U= null;
② put r into U;
③ for each query r_m in $RSet$ /* 依次扫描 $RSet$ 中待匿名的查询* /
④ if $|r.T_{exp} - r_m.T_{exp}| > \delta_T$ /* 判断时效近似性* /
⑤ get the next query in $RSet$;
⑥ else
⑦ $BoundaryObjectsComputing(r_m, btq, U)$; /* 计算边界对象参见第 5.1 节* /
⑧ if(δ_t-$DistortionDetection(r_m, btq, U)$) /* 判断 δ_t-质量模型参见第 5.2 节* /
⑨ dis= compute $SimDis(r_m, U)$ from t_s to $maxT$; /* 计算相似距离* /
⑩ if($mindis > dis$)
⑪ $mindis = dis$;
⑫ $r_{min} = r$;
⑬ endif
⑭ endif
⑮ endif
⑯ insert r_{min} into U ; /* 把查询 r 与具有最小相似距离的集合合并* /
⑰ repeat Step ③ to Step ⑫ until $|U|$ doesn't change;
⑱ if($|U| \geqslant k$)
⑲ δ_r- privacy detection in Section 5.3; /* 判断 δ_p-隐私模型* /
⑳ else
㉑ insert r into $RSet$; /* 插入待匿名对象集合* /
㉒ endif
㉓ endfor

5.1 边界对象的检测

如上所述, 匿名框的边界对象随着对象的运动而变化. 虽然在一个候选匿名集中所有移动对象的状态(初始位置和运动速度)已知, 但是在一段时间内, 追踪所有对象的运动进而获得所有时刻的边界对象是不现实的, 代价很昂贵.

实际上, 计算边界对象没必要追踪每一个对象的运动, 只需要关注在时刻 t 的边界以及比边界对象运动速度快的对象. 图 7 给出了一维情况下在 x 轴上运动的例子. 从时刻 t_i 到时刻 t, 任意一个对象在 x 轴的位置可以通过式(3)确定:

$$x = x_{t_i} + v_x \times (t - t_i). \tag{3}$$

通过解线性方程组可以获得图 7 中不同移动对象相遇的时刻与位置(即交叉点). 并且, 不是所有的交叉点都需要计算, 如图 7 中交叉点 A 对边界没有影响, 可以忽略. 所以只需要关注那些比边界对象运动速度快的查询, 同时通过式(3)解线性方程组计算出边界更换的时间和位置, 并记录在队列 BTQ 中, 以帮助 δ_t-质量模型和 δ_p-隐私模型的判断. 队列 BTQ 中包含的每一个对象形式为<$time$, $query$, $boundary$>, 其中 $time$ 表示边界对象更换的时间, $query$ 表示更换的边界对象标识, $boundary$ 表示是候选匿名框上/下/左/右哪一个边界.

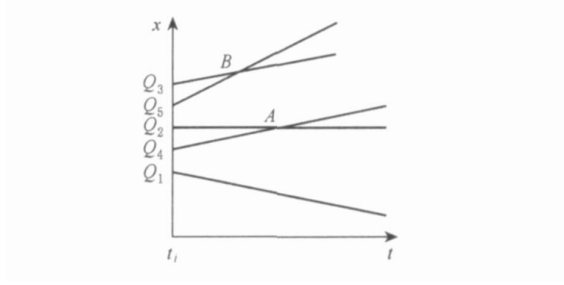

Fig. 7 Boundary objects detection.
图 7 边界对象检测

所以, 边界对象检测方法的主要思想是: 对于每一个候选匿名集, 针对 x 轴正方向, 寻找比当前边界速度大的对象, 根据式(3)解方程组计算边界更换时间; 针对 x 轴负方向寻找比当前边界速度小的对象, 计算边界更换时间. 根据对称性, y 轴方向上采用类似操作. 该算法较直观, 由于篇幅限制, 具体算法省略.

5.2 δ_t-质量模型检测

在第 5.1 节已计算出每一个候选匿名集的边界对象变更队列 BTQ, 结合该队列很容易获得候选匿名集边界对象及其位置. 对于 BTQ 中任意两个连续

时刻点 t_i, t_{i+1}, 设 $P_A = A_{height} + A_{width}$, $P_{L,t} = L_{x+,t} - L_{x-,t} + L_{y+,t} - L_{y-,t}$, $P_{v,t} = v_{x+,t} - v_{x-,t} + v_{y+,t} - v_{y-,t}$, 则根据 δ_q-质量模型的要求, 对任意时刻 $t \in [t_i, t_{i+1}]$,

$$\frac{1}{P_A}[P_{L,t_i} + P_{v,t_i}(t - t_i)] < \delta_q. \qquad (4)$$

计算不等式(4), 如果在$[t_i, t_{i+1}]$存在解, 则说明不满足 δ_q-质量模型. 具体见算法 2:

算法 **2.** δ_q-质量模型检测算法.

输入: time queue BTQ, queries set U, time t_s;

输出: false/true.

① $t_{i-1} = t_s$;

② while BTQ is not null/* 当BTQ不为空时* /

③ t_i = pop up the first time from BTQ;
/* 从BTQ中取出第1个元素赋值给 t_i* /

④ $t = \dfrac{P_A \sigma_q - P_L}{P_V} + t_i$;

⑤ if $t \geqslant t_i$ and $t < t_{i-1}$

⑥ return false; /* 式(4)无解* /

⑦ endif

⑧ $t_{i-1} = t_i$;

⑨ endwhile

⑩ return true.

5.3 δ_p-隐私模型检测

与第 5.2 节类似, 结合边界对象变更队列 BTQ, 可以检测候选匿名集是否满足 δ_p-隐私模型. 主要思想是对于任一个候选匿名集, 取出其 BTQ 中两个连续时刻t_i, t_{i+1}, 根据式(1)和式(2)计算候选匿名框的宽和高, 分别判断是否大于 $\Delta_p = \delta_p \times \min(A_{width}, A_{height})$. 若在候选匿名集的生命有效期内的任意一个连续时间段$[t_i, t_{i+1}]$, 两个不等式均无解, 同时候选匿名集大小大于 K, 则该候选匿名集可作为匿名结果成功返回. 反之, 若有其中任何一个不满足, 则把触发查询 r 插入查询待匿名集合 $RSet$. 具体算法参见算法 3:

算法 **3.** δ_p-隐私模型检测算法.

输入: time queue BTQ, queries set U, time t_s; /* 输入时间队列 BTQ、查询集合 U、时间 t* /

输出: false/true.

① $t_{i-1} = t_s$;

② while BTQ is not null/* 当BTQ不为空时* /

③ t_i = pop up the first time from BTQ
/* 设 $WB_t = L_{x+,t_i} - L_{x-,t_i}$, $HB_t = L_{y+,t_i} - L_{y-,t_i}$, $VWB = v_{x+,t} - v_{x-,t}$, $VHB_t = v_{y+,t} - v_{y-,t}$* /

④ $t_1 = \dfrac{\Delta_p - WB_{t_{i-1}}}{VWB_{t_{i-1}}} + t_{i-1}$; /* 判断从 t_{i-1} 到 t_i 是否每个时刻匿名框的宽均大于 δ_p* /

⑤ if $t_1 \geqslant t_{i-1}$ and $t_1 < t_i$

⑥ return false;

⑦ else

⑧ $t_2 = \dfrac{\Delta_p - HB_{t_{i-1}}}{VHB_{t_{i-1}}} + t_{i-1}$; /* 判断从 t_{i-1} 到 t_i 是否每个时刻匿名框的高均大于 δ_p* /

⑨ if $t_2 \geqslant t_{i-1}$ and $t_2 < t_i$

⑩ return false;

⑪ endif

⑫ endif

⑬ endwhile

⑭ return true.

每一个查询均有一个有效期, 如果该查询在有效期内没有匿名成功, 则从 $RSet$ 中去除该对象.

6 实验结果与分析

实验采用著名的 Thomas Brinkhoff[10] 路网数据生成器, 以城市 Oldenburg 的交通路网(周长大约 6000 km)作为输入, 生成模拟数据. 算法采用 Java 实现, 在处理器 P4 2.0 GHz、内存 2 GB 的平台上运行. 算法中的各参数默认值如表 1 所示:

Table 1　Default System Settings

表 **1**　实验参数及默认取值

Parameters	Default Values
Number of Queries	10000
Privacy Level(k)	5
δ_p	0.1% of the space
δ_q	1% of the min ($width$, $height$) of the space
δ_v/s	100

实验评测了匿名成功率、匿名时间、处理时间和平均匿名代价随着隐私度(k)增加的变化情况. 隐私度 k 的增加代表用户的隐私需求更加严格, 要求更多的用户匿名在一起从而保护用户隐私. 匿名成功率是成功获得匿名查询占查询提出总数的百分比. 如图 8 所示, 随着隐私度 k 的增长成功率逐渐下降. 但是, 即使隐私度增加到 $k = 8$, 成功率依然保持在 90%以上. 查询匿名时间指的是触发查询提出请求

到匿名成功的时间. 查询处理时间是指任何查询从提出到匿名成功的时间. 查询处理时间比匿名时间多了等待时间. 如图 9 所示, 无论是匿名时间还是处理时间, 均随着隐私度的增加而增长. 这是因为随着隐私度的增长, 每一个查询均需要更多的时间处理、等待才能匿名成功. 用每一个查询的匿名框的平均周长代表查询的平均匿名代价. 周长越长则查询处理代价越高. 如图 10 所示, 用户的匿名代价随着隐私度的增加而呈线性增长. 即随着隐私度的增加, 匿名框需要覆盖更多较远的对象从而满足隐私需求.

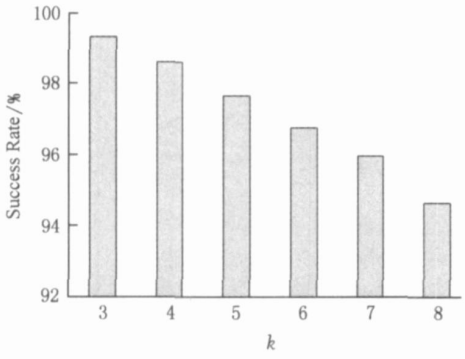

Fig. 8 Success rate.

图 8 匿名成功率

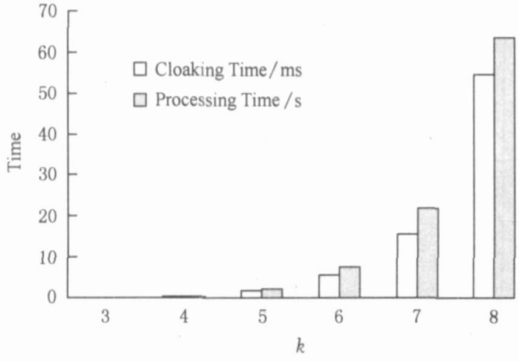

Fig. 9 Cloaking time and processing time.

图 9 匿名时间和处理时间

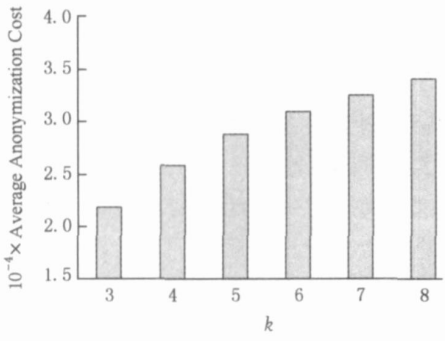

Fig. 10 Average anonymization cost.

图 10 平均匿名代价

7 结 论

本文研究了基于位置服务中连续查询的隐私保护问题. 阐述了现有的静态匿名算法不适用于连续查询隐私保护, 产生查询隐私泄露、匿名服务器工作代价大等问题. 并提出了 δ_t-隐私模型和 δ_q-质量模型可以有效地均衡隐私保护与服务质量, 在默认设置下, 基于该模型的贪心匿名算法可以在 2 ms 内匿名成功, 匿名成功率可达到 98%.

参 考 文 献

[1] Pan Xiao, Xiao Zhen, Meng Xiaofeng. Survey of location privacy-preserving [J]. Journal of Frontiers of Computer Science and Technology, 2007, 1(3): 268-281 (in Chinese) (潘晓, 肖珍, 孟小峰. 位置隐私研究综述[J]. 计算机科学与探索, 2007, 1(3): 268-281)

[2] Mokbel M F, Chow C Y, Aref W G. The new Casper: Query processing for location services without compromising privacy [C] // Proc of the 32nd Int Conf on Very Large Data Bases (VLDB). New York: ACM, 2006: 763-774

[3] Chow C, Mokbel M F. Enabling privacy continuous queries for revealed user locations [C] // LNCS 4605: Proc of the Int Symp on Advances in Spatial and Temporal Databases (SSTD). Berlin: Springer, 2007

[4] Gruteser M, Grunwald D. Anonymous usage of location-based services through spatial and temporal cloaking [C] // Proc of the Int Conf on Mobile Systems, Applications, and Services (MobiSys). New York: ACM, 2003: 163-168

[5] Xiao Zhen, Xu Jianliang, Meng Xiaofeng. P-sensitivity: A semantic privacy-protection model for location-based services [C] // Proc of the 2nd Int Workshop on Privacy-Aware Location-Based Mobile Services (PALMS). Piscataway, NJ: IEEE, 2008: 47-54

[6] Bamba B, Liu L. Supporting anonymous location queries in mobile environments with privacy grid [C] // Proc of Int Conf on World Wide Web (WWW). New York: ACM, 2008: 237-246

[7] Kido H, Yanagisawa Y, Satoh T. Protection of location privacy using dummies for location-based services [C] // Proc of the 26th Int Conf on the Physics of Semiconductors (ICPS). Piscataway, NJ: IEEE, 2005: 1248-1248

[8] Ghinita G, Kalnis P, Khoshgozaran A, et al. Private queries in location based services: Anonymizers are not necessary [C] // Proc of ACM SIGMOD 2008. New York: ACM, 2008

[9] Xu T, Cai Y. Location anonymity in continuous location based services [C] //Proc of Int Symp on Advances in Geographic Information Systems(GIS). New York: ACM, 2007

[10] Brinkhoff T. A framework for generating network-based moving objects [J]. An Int Journal on Advances of Computer Science for Geographic Information Systems (GeoInformatica), 2002, 6(2): 153-180

Pan Xiao, born in 1981. PhD candidate at Renmin University of China. Her main research interests focus on mobile data management.

潘　晓, 1981 年生, 博士研究生, 主要研究方向为移动数据管理.

Hao Xing, born in 1985. Master candidate at Renmin University of China. Her main research interest focuses on mobile data management.

郝　兴, 1985 年生, 硕士研究生, 主要研究方向为移动数据管理(haoxing@ruc.edu.cn).

Meng Xiaofeng, born in 1964. Professor and PhD supervisor, Secretary General of Dadabase Society of China Computer Federation. His main research interests include Web data integration, XML database, and mobile data management.

孟小峰, 1964 年生, 教授, 博士生导师, 中国计算机学会数据库专委会秘书长, 主要研究方向为 Web 数据管理、XML 数据库、移动数据管理(xfmeng@ruc.edu.cn).

Research Background

With the blooming of sensor and wireless mobile devices, it is easy to access mobile users' location anytime and anywhere. On one hand, location based services are more and more valuable and important. On the other hand, privacy issues, including location privacy and query privacy, in location based services raised by such applications, have attracted more and more attention. In this paper, we consider query privacy preserving for snapshot and continuous queries. To address this issue, we firstly propose δ_p-privacy model and δ_q-distortion model to balance the tradeoff between privacy preserving and quality of services, and use the perimeter of cloaking region to measure the distortion of the location information. Then, the location distortion is mapped to the temporal similar distance between two queries. Finally, a greedy algorithm is proposed to find cloaking set for snapshot and continuous queries. Average cloaking success rate, cloaking time, processing time and anonymization cost for successful requests are evaluated with increasing privacy level. Experimental results validate the efficiency and effectiveness of our proposed algorithm. This research is partially supported by the National Natural Science Foundation of China(Nos. 60833005, 60573091), the National 863 High-Tech Research and Development Plan of China (Nos. 2007AA01Z155, 2009AA011904).