计算机研究与发展
Journal of Computer Research and Development
DOI:10.7544/issn1000-1239.2015.20140532
52(11):2628-2644,2015

路网环境下保护 LBS 位置隐私的连续 *K*NN 查询方法

周长利[1,2]　马春光[1]　杨松涛[1]
[1](哈尔滨工程大学计算机科学与技术学院　哈尔滨　150001)
[2](华侨大学计算机科学与技术学院　福建厦门　361021)
(machunguang@hrbeu.edu.cn)

Location Privacy-Preserving Method for LBS Continuous *K*NN Query in Road Networks

Zhou Changli[1,2], Ma Chunguang[1], and Yang Songtao[1]
[1](*College of Computer Science and Technology, Harbin Engineering University, Harbin* 150001)
[2](*College of Computer Science, Huaqiao University, Xiamen, Fujian* 361021)

Abstract　Location privacy preservation and query service quality are a pair of contradiction in location based service (LBS). In the road network, there are a lot of limiting factors to be considered for continuous query. How to protect location privacy efficiently and acquire accurate continuous query results of places of interest (POIs) are great challenges in the road network. In this paper, based on the idea of using fake location, a query algorithm is proposed firstly, which picks the intersections of the road network gradually to form an anchor sequence to query POIs, and the query algorithm can not only achieve location privacy preservation but also deduce accurate K nearest neighbor (KNN) query results. And then, based on the idea of sending fake queries and constructing query anonymity group, a trajectory privacy preservation algorithm is proposed, which is used to resist continuous query correlation attack and movement model inference attack. At last, a discussion about the trade-off between privacy preservation and query service quality is given in the road networking LBS. The performance analysis and experiments show that our methods provide strong location privacy preservation and get accurate query results in the road network, and our algorithms have favorable timeliness and well-balanced data communication overhead.

Key words　location privacy-preserving; continuous query; location based service (LBS); road network; correlation attack

摘　要　位置隐私保护与基于位置的服务(location based service, LBS)的查询服务质量是一对矛盾,在连续查询(continuous query)和实际路网环境下,位置隐私保护问题需考虑更多限制因素.如何在路网连续查询过程中有效保护用户位置隐私的同时获取精确的兴趣点(place of interest, POI)查询结果是目前的研究热点.利用假位置的思想,提出了路网环境下以交叉路口作为锚点的连续查询算法,在保护位置隐私的同时获取精确的 K 邻近查询(K nearest neighbor, KNN)结果;基于注入假查询和构造查询匿名组的方法,提出了抗查询内容关联攻击和抗运动模式推断攻击的轨迹隐私保护方法,并在分析中给出了位置隐私保护和查询服务质量平衡方法的讨论.性能分析及实验表明,该方法能够在连续查询中提供较强的位置隐私保护,并具有良好的实效性和均衡的数据通信量.

收稿日期:2014-06-16;**修回日期**:2015-02-04
基金项目:国家自然科学基金项目(61170241,61472097);教育部高等学校博士学科点专项科研基金项目(20132304110017);黑龙江省杰出青年基金项目(JC201117);黑龙江省教育厅科学技术研究项目(12541788)

关键词 位置隐私保护;连续查询;基于位置的服务;路网;关联攻击

中图法分类号 TP309.2

基于位置的服务(location based service, LBS)为人们生活带来方便的同时,也带来了位置隐私泄露问题.一方面,带有位置感知功能的移动智能终端使 LBS 得到广泛应用,这些服务包括基于位置的兴趣点查询、基于位置的社交网络及触发式广告优惠券推送服务等,这些服务为人们生活提供了更多便利快捷;另一方面,LBS 需要用户提供精确的实时位置数据作为获取相应服务的必要前提,由于位置本身所具有的时空关联特性,位置数据的泄露会间接带来其他隐私信息(如身份、兴趣爱好、工作单位等)被推断的可能,因此,位置隐私保护一直受到学术界和商业界的广泛研究和关注.

LBS 中的兴趣点(place of interest, POI)查询服务可以分为 K 邻近查询(K nearest neighbor, KNN)和 R 范围查询(R range query),如"查找距离我最近的 K 个川菜馆"和"查找距离我 R(单位 km)范围内的所有电影院".而按照查询方式可以分为快照查询(snapshot query)和连续查询(continuous query),快照查询是指用户发送 1 次兴趣点查询请求,而连续查询是指连续发送多次快照查询,如"5 min 内持续报告距离我最近的 K 个川菜馆".在 LBS 兴趣点查询服务中,如何有效隐匿用户真实位置,并高效获取查询服务是 LBS 位置隐私保护追求的目标.

LBS 中的位置隐私保护技术主要包括位置 k 匿名技术和假位置技术 2 类:

1) 位置 k 匿名技术.是将用户与其他 $k-1$ 个位置上的用户构成匿名组,降低用户真实位置分辨率.典型的技术是构建 k 匿名框(cloaking region, CR)[1].该方法构造包含 k 个用户的匿名框,并将匿名框替代用户真实位置提交给 LBS 服务器来查询兴趣点,使攻击者无法区分匿名框中的 k 个用户.然而这种方法在连续查询过程中存在暴露用户隐私的风险[2-3].如图 1 实线匿名框所示,为用户连续构造的 3 个匿名框的交集为 A,则可推断利用这 3 个匿名框发出查询的均是用户 A,其余用户为参与者,用户 A 的查询请求暴露.针对此问题,文献[4-5]提出了解决方法,基本思想是维护初始匿名框中的绝大多数用户在接下来的匿名框中继续存在,图 1 虚线框所示.因此,连续匿名框的构造需要确保初始匿名框内用户的运动方式具备一定的一致性,但在用户处于同一条路段的情况下,即使连续匿名成功,用户的轨迹隐私也会泄露,此类方法仅适用于欧氏空间而没有考虑实际中的路段多样性[6].同时,在实际路网中总是维护初始绝大多数用户仍在连续匿名框中是困难的.同时,受路径限制等影响,隐私保护问题研究需考虑更多实际的受限因素.

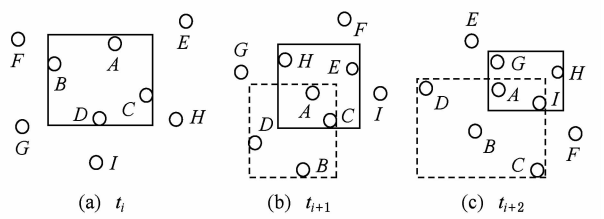


Fig. 1 Cloaking regions for continuous query.

图 1 匿名框连续查询

2) 假位置技术.是利用假位置[7]替代用户真实位置发送兴趣点查询.然而,假位置存在查询结果不够精确并给 LBS 服务器带来不必要的额外负担等问题.针对这个问题,文献[8]提出了利用假位置(锚点)的精确查询方法 SpaceTwist,该方法利用锚点发送查询请求,并根据 LBS 服务器返回的兴趣点查询候选集并结合自身真实位置计算所需的精确查询结果,LBS 服务器负担相对较小,但该方法只适用于快照查询,并且是基于欧氏空间的 KNN 查询,在路网环境下该方法并不适用.

本文借鉴 SpaceTwist 中利用锚点查询的方法,在实际路网环境下设计无需构造匿名框的连续查询方法.该方法利用多跳交叉路口作为连续查询锚点,获取精确 KNN 查询结果并能够保护用户位置隐私,同时利用注入假查询和构造匿名查询组的方法,解决了由于连续查询内容关联带来的轨迹隐私泄露问题和用户运动模式分析带来的推断攻击问题.在性能分析中,本文给出了隐私保护与查询质量平衡方法的讨论.

1 相关工作

LBS 中的位置隐私泄露问题直接影响了各种相关服务的推广和应用.位置隐私保护方法可以通过构造匿名框和利用假位置的方法实现.针对快照查询时位置匿名方法的不足,文献[9]形式化分析了静态匿名方法在连续查询中失效的问题.连续匿名

方法是近年来的研究重点[4-5,10-11],其基本思想是确保最初构造匿名框中的绝大多数用户继续在接下来多个连续查询匿名框中,以避免推断攻击. 文献[4]利用网格为用户生成连续匿名框并利用融合的方法来解决面对大量查询时的查询瓶颈问题. 文献[5]考虑了不同用户运动方式的相似性来构造连续查询匿名组. 文献[10]以匿名框长和宽的变化来衡量不同用户运动状态是否具有相似性,从而构造合适的匿名框. 文献[11]通过维护 1 个最大匿名团的方法,来保证连续查询过程中的匿名有效性.

然而,在连续查询过程中,一些关联推断因素结合特定背景可能会带来新的攻击. Mix-zone[12]是最早被提出抗关联攻击的方法,然而在用户更换假名并离开 Mix-zone 后如果继续发起原有查询请求,攻击者可利用相同的查询作为关联进出 Mix-zone 用户的条件,因此该方法无法抵抗连续查询内容关联攻击[13]. 文献[14]指出了在连续查询中还需要保证查询内容的多样性. 在连续查询过程中,运动方式推断也是常见的攻击方法,其中最大速度攻击[15]是利用用户连续查询中前后发布的 2 个匿名框及相应重叠部分来推断用户准确位置,文献[16]给出了抵抗重叠区域攻击的一类方法. 文献[17]给出了一种连续攻击的算法,并给出了匿名性的度量方法.

在利用假位置方面,SpaceTwist[8]方法利用随机选取的锚点位置替代用户真实位置查询 KNN 兴趣点,并根据返回的候选集计算出精确结果,该方法能够有效保护用户位置隐私并保证较好的服务质量;然而该方法只适用于快照查询并且在欧氏空间下查询兴趣点集. 文献[18-19]在不同架构下对 SpaceTwist 进行了改进. 本文将借鉴 SpaceTwist 利用锚点的思想,来解决路网环境下连续查询中的位置隐私泄露问题.

本文研究内容及贡献如下:

1) 针对位置隐私保护研究多基于欧氏空间的不足,提出了路网环境下连续 KNN 查询方法. 该方法以路段交叉点为连续查询锚点,将路段上所有用户的连续兴趣点查询转化为等价的必达路段顶点查询,在保证路段上所有用户位置隐私的同时,实现用户 KNN 兴趣点的精确查询.

2) 针对连续查询中存在的查询内容关联攻击和运动模式推断攻击带来的轨迹隐私泄露问题,本文基于路网多样性提出了注入假查询方法和构造查询匿名组的方法来抵抗上述攻击.

3) 性能分析给出了位置隐私保护和查询质量平衡方法的讨论,实验表明本方法具有良好的实效性和较低的数据通信量.

本文首次利用路网顶点作为锚点并利用锚点来解决非欧氏空间下路网兴趣点连续 KNN 查询的位置隐私保护.

2 系统架构

位置隐私保护架构主要分为 2 类:2 层架构和 3 层架构[20-21]. 2 层架构中实体主要包括用户和 LBS 服务器,用户端执行位置隐私保护方法,并将处理后的位置信息提交给 LBS 服务器,这种架构简单,但用户端负担较重;3 层架构是在 2 层架构实体之间加入可信位置数据处理服务器,该服务器将泛化处理后的位置发送给 LBS 服务器,保证位置隐私安全,并根据 LBS 服务器返回的候选查询结果计算出精确的查询结果返回给用户. 3 层架构能够有效降低用户负担,提供更强的位置隐私保护,但需要用户提供真实位置信息. 本文采用 3 层架构,如图 2 所示:

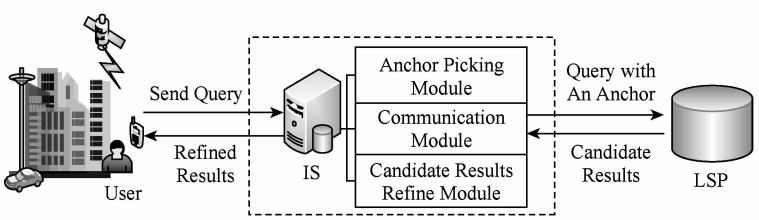


Fig. 2 System architecture.

图 2 系统架构图

定义 1. 架构内实体. 架构内的 3 类实体可以表示为($User$, E^{IS}, E^{LSP}),$User$ 表示带有智能终端的用户集合,E^{IS}表示中间服务器(intermediate server, IS)集合,E^{LSP}表示基于位置的服务提供商(LBS provider, LSP)(即 LBS 服务器)集合.

每个用户 $u_k \in User$ 配备具有定位功能及速度矢量感知功能的智能终端,每个中间服务器由 3 个模块构成:锚点计算模块负责根据用户提供真实位

置并结合用户运动状态计算出锚点,通信模块代理用户利用锚点向 LBS 服务器发起兴趣点查询,LBS 服务器以增量查询方式逐步返回候选查询结果,查询求精模块根据候选结果计算用户所需精确兴趣点并返回给用户.

一定数量的中间服务器被部署在路网中,用户将连续查询请求发送给他的 1 个邻近的中间服务器 $E_i^{IS}\in E^{IS}$,E_i^{IS} 掌握其管理区域内路网分布情况并为用户选择连续查询锚点.

定义 2. 连续查询. 用户对兴趣点连续查询请求可表示为 $Q_{u_k}=(u_k,loc,\boldsymbol{v},des,Vel,C,T)$,其中,$loc$ 表示用户当前真实位置,$\boldsymbol{v}$ 表示当前用户速度向量,des 表示用户目的地,Vel 表示用户历史速度向量集合,C 表示用户查询内容,T 表示查询有效期;中间服务器代理用户利用锚点发起的某个快照查询可以表示为 $Q_{IS}=(E_i^{IS},loc_{anchor},C)$,其中 E_i^{IS} 表示某个中间服务器的唯一标识,loc_{anchor} 表示锚点位置,查询有效期 T 内同一查询内容 C 的多次快照查询构成该用户 u_k 的连续查询,连续查询中连续锚点构成锚点序列.

所谓轨迹,是同一用户在时间轴上连续的位置序列. 轨迹隐私,则是将某个连续位置序列映射到唯一的用户身份标识上,因此轨迹隐私泄露多指攻击者找到连续位置序列与用户唯一身份标识间的映射关系.

定义 3. 位置关联攻击. 如果 $\exists$ 映射 f 使得 $f:[(ConItem,\varepsilon),(loc_1,loc_2,\cdots,loc_n),Context]\rightarrow u_k$ 成立,则称用户受到位置关联攻击,其中 $ConItem$ 和 ε 分别表示关联项及关联强度,$Context$ 为背景知识等输入.

由此可以看出,攻击能否实现取决于攻击者对 f 的构造能力,攻击者在选取 $ConItem$ 时,可以选择如查询内容 C 等作为位置序列的关联因素;ε 表示在 $ConItem$ 作为关联因素的条件下,位置序列 $(loc_1,loc_2,\cdots,loc_n)$ 的关联强度,ε 取值越大,位置间的关联程度越高.

本文提出的由查询内容关联带来用户的轨迹隐私泄露则可以表示为

$$f:[(C_i,\varepsilon_{c_i}),(loc_{anchor1},loc_{anchor2},\cdots,loc_{anchorn}),Context]\rightarrow u_k, \quad (1)$$

其中,ε_{c_i} 强度由查询内容 C_i 一致性等因素决定,利用每一个位置 $loc_{anchor1}$ 发出的查询内容为相同查询请求,则 ε_{c_i} 值越高连续位置关联度越高,攻击者构造出 f 的可能性越高.

同理,本文提出的运动模式相似性带来的轨迹推断攻击可以表示为

$$f:[(Vel,\varepsilon_{Vel}),(loc_1,loc_2,\cdots,loc_n),Context]\rightarrow u_k, \quad (2)$$

其中,ε_{Vel} 强度由速度向量集合 Vel 一致性等因素决定,此时 $Context$ 可以包含攻击者通过观察获得的用户停留行进知识,位置序列速度一致性越高,连续位置关联度越高,攻击者构造出 f 的可能性越高.

网络中恶意攻击者的能力包括掌握路网连通情况、用户在路网的分布情况,并具有一定的关联分析能力. 同时,由于 LBS 服务器存在泄露用户隐私的可能,我们将 LBS 服务器看作半可信的实体.

3 连续查询位置隐私保护方法

本节主要分为 3 部分,首先基于路网有向图定义给出移动用户路网位置的预测方法;然后基于预测方法,给出保护位置隐私的连续查询算法;最后在连续查询算法的基础上针对查询请求关联攻击和运动模式推断攻击设计轨迹隐私保护方法.

3.1 路网描述及移动用户位置预测

行驶在路段上的用户不断发起 LBS 连续查询请求,对比利用连续多个匿名框发起的每次快照查询方法,本文连续查询中的每次快照查询不利用匿名框,而是利用锚点序列,每个锚点由 1 个位置坐标表示. 利用锚点无需 LBS 服务器具备处理非精确位置(如匿名框)查询的能力. 因此,预测用户行驶位置,并为用户选择合适锚点序列是确保连续查询成功的关键.

在路网受限空间中,行驶在路网中的用户受路网限制,行驶方向与行驶路线受路网影响,另一方面这种限制也使得用户的行驶状态具备一定的确定性和可预知性. 一般情况下,用户总是选择路径短、路况好的路段构成到达目的地的行驶路径,而路径中的绝大多数路段的方向总是大致指向目的地(否则用户无法到达目的地),这为用户行驶路径的预测提供了依据.

路网可以抽象成有向图结构,如图 3 所示. 我们定义路网相关概念如下:

定义 4. 路网图. 路网结构有向图可以用 2 元组 $digraph=(V,E)$ 表示,其中 V 是图中顶点集合,$v_i\in V$ 表示路网交叉路段节点;E 表示图中有向边集合,由于路段有向性,任意 1 条有向边可以用顶点有序对表示(v_m,v_n),该路段表示顶点 v_m 的 1 条出边,同时也表示 v_n 的 1 条入边.

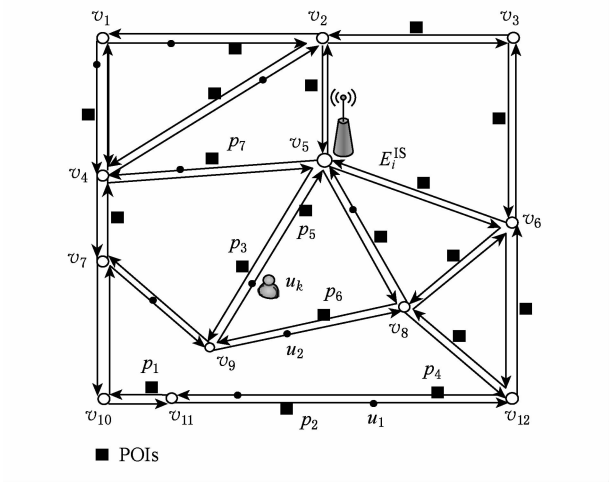


Fig. 3 Road networks graph.

图 3 路网图

在图 3 中,$v_1 \sim v_{12}$ 均为路段顶点,如 $(v_9, v_5) \in E$ 表示 1 条由 v_9 到 v_5 的路段,即顶点 v_9 的 1 条出边.路段上分布着固定及移动点,如用户 u_k、兴趣点 p_i 等.如图 4 所示,$u_k \in (v_9, v_5)$ 表示位于路段 (v_9, v_5) 上的用户,路段指向同时表示用户的行进方向,也表示了兴趣点位于 2 点间路段某一侧,如 $p_3 \in (v_5, v_9)$ 与 $p_5 \in (v_9, v_5)$ 表示分别位于用户 u_k 行驶方向左右两侧的兴趣点.

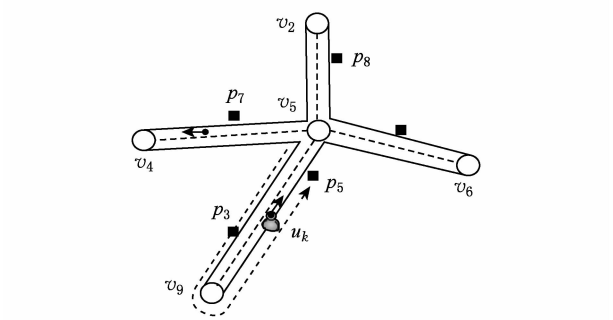


Fig. 4 Directed road network distance.

图 4 有向路网距离图

定义 5. 路网距离.边上任意点 v_i 到点 p_i 的行驶路径长度称为路网距离,用 $dist_r(v_i, p_i)$ 表示,通常情况下 $dist_r(v_i, p_i)$ 表示顶点 v_i 到点 p_i 的最短行驶路径.

由于路网空间限制,$dist_r(v_i, p_i)$ 与 $dist_r(p_i, v_i)$ 可能不相等,且欧氏距离近的点可能路网距离较远,由顶点 v_5 出发到兴趣点 p_3 的路网距离为 $dist_r(v_5, p_3)$,如图 4 所示,顶点 v_5 到兴趣点 p_5 的路网距离可以表示为 $dist_r(v_5, p_5) = dist_r(v_5, v_9) + dist_r(v_9, p_5)$,而顶点 v_5 到兴趣点 p_5 的欧氏距离比到兴趣点 p_3 更小.

中间服务器部署后,每个 $E_i^{IS} \in E^{IS}$ 掌握负责区域内路网顶点、路段连通及用户分布情况,定义如下:

定义 6. E_i^{IS} 路段表.E_i^{IS} 掌握负责区域内路网顶点集合 $V_{E_i^{IS}} \subseteq V$ 及每个顶点坐标,每个顶点的所有连通路径可以用其邻近的 1 跳顶点描述$\langle v_i \in V_{E_i^{IS}} | \{v_m, v_n, v_k, \cdots\} \rangle$,其中 v_i 表示区域内任意顶点,$\{v_m, v_n, v_k, \cdots\}$ 表示与 v_i 直接联通的所有邻近顶点集合,该区内全部顶点联通情况描述构成 E_i^{IS} 管理区域内的 2 维路段表.

图 3 表示 E_i^{IS} 管理的路网区域,顶点 v_5 的所有 1 跳连通顶点描述为$\langle v_5 \in V_{E_i^{IS}} | \{v_2, v_4, v_9, v_8, v_6\} \rangle$,则 (v_5, v_9) 表示该区域内 1 条连通路径,对于 E_i^{IS} 来说,掌握区域内每个顶点的连通情况就可以掌握路网情况.

用户 u_k 首先向最近的某个 E_i^{IS} 发送连续查询请求$(u_k, loc, \boldsymbol{v}, des, Vel, C, T)$,$E_i^{IS}$ 根据其位置信息利用反向地理编码技术确定用户所在的路段.由于路段的限制,行驶在某一路段上的用户如果目的地不在该路段,则其到达该路段指向顶点的可能性接近 100%,如用户 $u_k \in (v_9, v_5)$,E_i^{IS} 而很容易根据用户速度向量 $\boldsymbol{v}$ 确定用户即将达到的顶点 v_5,如图 4 所示,而 u_k 达到 v_5 后可能选择 v_2 作为下一个到达的路网顶点.因此,用户的行驶路径预测可以转换为用户当前所在路段必达顶点的下一路网顶点选择问题.

当用户到达当前路段顶点后,影响用户选择下一目标顶点的因素之一为用户目的地 des 位置坐标,应选择路段指向与目的地方向偏离较小的下一目标顶点;同时,用户的历史速度向量绝大多数指向目的地方向,即用户选择的历史路段总是与目的地方向偏差不大,因此用户的历史速度向量与下一目标顶点夹角越小,意味着用户选择该顶点作为下一目标顶点的路段与历史速度向量集不会有较大偏差,即选择该路段的行进方向也是向着目的地方向前进的,如图 5 所示.我们主要考虑用户目的地方向及与历史行进向量一致性 2 个因素来判断用户选择下一跳顶点.即将到达路口 v_i 的用户可选择的下一跳顶点有多个,如图 3 中 u_k 达到 v_5 后可能选择的下一跳顶点集合为 $\{v_4, v_2, v_6, v_8, v_9\}$,对于 v_i 的每一个下一跳顶点 v_j 均执行式(3),计算权值 Rv_j:

$$R_{v_j} = \delta_1 \times \theta((v_i, v_j), (v_i, des)) + \delta_2 \times \sum_{m=1}^{w} \theta((v_i, v_j), \boldsymbol{v}_m)/w, \quad (3)$$

其中,δ_1 和 δ_2 为影响系数,可分别取 0.6 和 0.4;θ 为两个向量夹角;$\boldsymbol{v}_m$ 为用户历史速度向量,$\boldsymbol{v}_m \in Vel$. 由式(3)可以看出,Rv_j 越小,v_j 作为下一跳目标顶点与目的地偏差越小,选择该顶点到达目的地可能性越高.

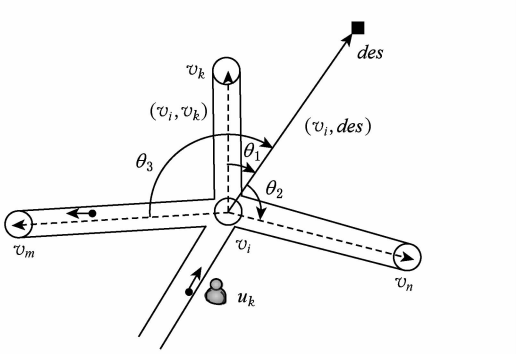


Fig. 5 Next-hop vertex prediction.

图 5 下一跳顶点判断

然而,式(3)虽然能够对路段进行初筛,选出方向大致指向目的地的若干条路段,但如何在这若干条路段中选择最好的路段仍需根据实际路况. 考虑到路段距离和路况等实时因素,在式(3)筛选出的路段集合$\{Rv_j \mid Rv_j \leqslant T_\lambda\}$中($T_\lambda$ 为阈值),E_i^{IS} 为每一个 Rv_j 计算权值 Qv_j,如式(4)所示:

$$Qv_j = \mu_1 \times dist_{\mathrm{r}}(v_i, v_j) + \mu_2 \times cap_{\mathrm{r}}(v_i, v_j), \quad (4)$$

其中,μ_1 和 μ_2 为影响系数,分别取 0.3 和 0.7;$cap_{\mathrm{r}}(v_i, v_j)$为路段$(v_i, v_j)$上移动用户数,用来描述路段拥堵状况. 显然,$Qv_j$ 值越小,该路段被选择的概率越高.

这样我们将用户行驶路径预测转换成对用户下一跳路网顶点选择问题,通过叠加目的地位置、历史轨迹、路段距离和路况等因素,预测用户最可能选择的下一路段. 中间服务器端用户下一顶点预测算法如下:

算法 1. 用户行进下一路网顶点预测算法.

① procedure:E_i^{IS} 收到用户 u_k 查询请求(u_k, loc_{u_k}, $\boldsymbol{v}$, des, Vel, C, T),构造小顶堆 W;

② E_i^{IS} 利用 loc_{u_k} 计算用户所在路段 $u_k \in (v_m, v_n)$;

③ for 每个 v_n 的 1 跳直接连接节点 v_i do

④ 根据式(3)计算 Rv_i;

⑤ while $Rv_i \leqslant T_\lambda$ do

⑥ 利用式(4)计算 Q_{v_i};

⑦ $W \leftarrow v_i$;

⑧ end while

⑨ end for

⑩ return $v_i \in W$; /* 取 W 中的首个 v_i 作为预测结果 */

⑪ end procedure

3.2 连续查询锚点选择及查询算法

在文献[8]中,锚点根据用户欧氏空间位置随机选择,利用锚点查询的兴趣点候选集在求精过程中,KNN 兴趣点也是根据用户真实位置与兴趣点的欧氏距离筛选出来. 实际路网环境下,用户 KNN 兴趣点不再以欧氏距离来计算而是以用户与兴趣点的路网距离来计算,由于缺少路网分布知识,用户也很难仅根据兴趣点的坐标判断最近的兴趣点,因此 SpaceTwist 不适用于路网环境.

基于兴趣点的路径可达特性,我们将所有兴趣点看作分布在不同路段上的点,如图 6 不同形状的实心点代表不同的兴趣点. 路网中以顶点为起点的 KNN 查询可以以该点为起点对路网进行广度搜索遍历获取查询结果. 而由于路段限制,当用户查询到位于其他路段上的某个兴趣点后,用户需要先到达可以转向的路口才能抵达兴趣点路段,而用户的行驶方向决定用户必须到达所在路段指向路口顶点后才能变换路段(否则违章驾驶). 如图 6 所示,u_k 要到达兴趣点 p_7 必须先要到达路口 v_5.

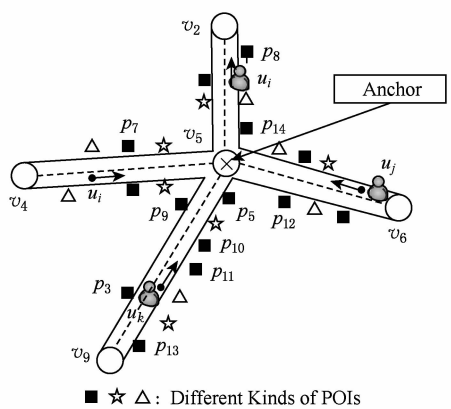


Fig. 6 Distribution of POIs.

图 6 兴趣点分布

因此,我们得到如下定理:

定理 1. 处于路段(v_m, v_n)任意位置的任意用户 u_k,除路段(u_k, v_n)上的兴趣点外,以必达顶点 v_n 为查询起点所得路网 KNN 兴趣点集 $P_{\mathrm{KNN}}^{v_n}$ 与以该用户 u_k 为查询起点所得路网 KNN 兴趣点集 $P_{\mathrm{KNN}}^{u_k}$ 相同.

证明. 如图 6 所示,用户 u_k 行驶在路段(v_9, v_5)上,不考虑用户与即将到达顶点间路段(u_k, v_5)上的兴趣点,设由顶点 v_5 作为查询起点所得路网环境下

的 4NN 兴趣点集为 $P_{4NN}^{v_5}=\{p_{14},p_9,p_{12},p_7\}$，$\forall p_i\in P_{4NN}^{v_5}$均有 $dist_r(v_5,p_i)<dist_r(v_5,p_j)$成立，其中 $p_j\notin P_{4NN}^{v_5}$. 因为$\forall u_k\in(v_9,v_5)$到顶点 v_5 的路网距离 $A=dist_r(u_k,v_5)$在某一时刻均为定值，所以对于任意 $p_i\in P_{4NN}^{v_5}$均有 $A+dist_r(v_5,p_i)<A+dist_r(v_5,p_j)$成立，即 $dist_r(u_k,p_i)<dist_r(u_k,p_j)$成立，因此，在不考虑路段$(u_k,v_n)$上兴趣点的情况下，$\forall u_k\in(v_9,v_5)$的 KNN 兴趣点集 $P_{KNN}^{u_k}=P_{KNN}^{v_5}$. 证毕.

定理 1 并未考虑用户行驶路段前方的兴趣点，如图 6 中 u_k 前方的兴趣点 p_5,p_{10},p_{11}，这是由于路段行驶方向限制，处于该路段的兴趣点总是与 v_5 距离较远，如 v_5 到 p_5 的路网距离(图 4 虚线所示). 因此，处于(u_k,v_5)的兴趣点很难被选为 v_5 的 KNN 兴趣点，但这类点由于处于用户行进方向上，却是用户路网距离较近的点即用户可能立即到达的兴趣点. 综上 2 种情况，行驶在路段(v_9,v_5)上用户 u_k 的 KNN 兴趣点集求解包含 2 个过程：1)获取路段(u_k,v_5)上的兴趣点集 Pu_kv_5；2)获取以 v_5 为查询起点的 XNN 兴趣点集 $P_{XNN}^{v_5}$. 其中 $X=K-|Pu_kv_5|$. 最后合并 2 个查询集 $Pu_kv_5\cup P_{XNN}^{v_5}$，即为用户的 KNN 兴趣点查询结果集. 综上，用户路网 KNN 兴趣点查询定理如下：

定理 2. $\forall u_k\in(v_m,v_n)$的某一类兴趣点的 KNN 查询结果集 Pu_k 均可以用并集 $Pu_k=Pu_kv_n\cup P_{(K-\alpha)NN}^{v_n}$来表示，其中集合 Pu_kv_n 表示路段(u_k,v_n)上的查询兴趣点集，$P_{(K-\alpha)NN}^{v_n}$表示以顶点 v_n 为查询起点的$(K-\alpha)$最邻近兴趣点集合，α 为集合 Pu_kv_n 的势.

这样，行驶在任意路段(v_m,v_n)上的所有用户$\{u_i|i\geqslant 1,i\in\mathbb{N}_+\}$的兴趣点查询可以转换为用户前方路段$(u_i,v_n)$上的同类兴趣点查询和即将到达路网顶点 v_n 为锚点的$(K-\alpha)$最近邻兴趣点查询 2 步操作. 对于用户来说，兴趣点集合 $P_{(K-\alpha)NN}^{v_n}$的查询并非利用用户的真实位置，而是用户前进方向的必达顶点，本文将其看作锚点，而该查询结果集大小受集合 Pu_kv_n 的势 α 的影响. 因此，在连续查询中，首先获取查询集合 Pu_kv_n 及其势 α，然后根据 α 以 v_n 为锚点获取$(K-\alpha)$最邻近兴趣点集 $P_{(K-\alpha)NN}^{v_n}$，并计算用户连续查询中的首次 KNN 兴趣点快照查询结果集 Pu_{k1}；随着用户行进不断经过此类兴趣点，α 会逐渐减小(此处将已经过的兴趣点看作与用户路网距离较大点，因为在用户遵守交通法规的情况下需要到下一路口掉头后再访问该点)，$(K-\alpha)$逐渐增大，即以 v_n 为锚点获取$(K-\alpha)$最邻近兴趣点集 $P_{(K-\alpha)NN}^{v_n}$逐步增大.

当用户更新位置获取连续查询中的第 2 次快照查询时，则以逐步扩大的 $P_{(K-\alpha)NN}^{v_n}$集合来计算新的查询结果集 Pu_{k2}，确保连续查询结果的准确性. 当用户在路段(v_m,v_n)上时，不断更新自己的位置获取新的扩大 $P_{(K-\alpha)NN}^{v_n}$计算 Pu_{ki}，直至用户离开此路段. E_i^{IS} 根据式(3)和式(4)能为用户预测到达顶点 v_n 后的下一跳顶点 v_{n+1}，使 E_i^{IS} 能够利用定理 1 和定理 2 以 v_{n+1} 为新锚点继续执行该连续查询直至该查询结束. 中间服务器端用户路网兴趣点连续查询算法如下：

算法 2. 中间服务器端路网兴趣点连续查询算法.

① procedure：E_i^{IS} 收到用户 u_k 的兴趣点查询请求$(u_k,loc_{u_k},\boldsymbol{v},des,Vel,C_{u_k},T_{u_k})$，$E_i^{IS}$ 代理用户 u_k 发给 LBS 服务器的每条查询的格式为$(E_i^{IS},loc_{v_n},C_{u_k},T_i)$，生成数组 $Cache_{v_n}$，用来存储 v_n 所有入边路段上的全部查询兴趣点；

② 依据 loc_{u_k} 计算用户所在路段 $u_k\in(v_m,v_n)$；

③ if C_{u_k} 不在 $Cache_{v_n}$ 中 then

④ $Cache_{v_n}\leftarrow$向 LBS 服务器获取 v_n 的 l 条随机入边路段上的全部 C_{u_k} 查询兴趣点；

⑤ end if

⑥ while u_k 下一目的顶点为 v_n 且 loc_{u_k} 是 u_k 周期性连续报告的最新位置坐标 do

⑦ 依据 $Cache_{v_n}$ 计算 u_k 所在路段(u_k,v_n)上的 C_{u_k} 查询兴趣点集 Pu_kv_n，$\alpha=|Pu_kv_n|$；

⑧ if $\alpha<K$ then

⑨ 以 v_n 的位置为锚点发起 1 次$(K-\alpha)$NN 快照查询；

⑩ 获取 LBS 服务器返回的兴趣点查询结果集 $P_{(K-\alpha)NN}^{v_n}$；

⑪ else $P_{(K-\alpha)NN}^{v_n}=\varnothing$；

⑫ $Pu_{ki}=Pu_kv_n\cup P_{(K-\alpha)NN}^{v_n}$；

⑬ end if

⑭ return Pu_{ki}；

⑮ 获取 u_k 下一时间周期更新的位置坐标；

⑯ end while

⑰ 结束以 v_n 为锚点的 LBS 查询；

⑱ if $T_{now}-T_{rec}<T_{u_k}$ then /* 查询未完成 */

⑲ 为 u_k 构造匿名查询组；

⑳ end if

㉑ 返回第②行;

㉒ end procedure

算法 2 行⑥～⑰说明当用户未离开所在路段前,需要周期报告自己当前的位置,以便中间服务器根据其实际的行进情况不断返回其所查的 KNN 兴趣点集合. 在 2 个兴趣点集 $P^{v_n}_{(K-\alpha)\mathrm{NN}}$ 和 Pu_kv_n 的查询过程中,$P^{v_n}_{(K-\alpha)\mathrm{NN}}$ 总是以用户行进指向的 v_n 为锚点进行查询的,因此用户可能处于 v_n 的多个入边路段上,这种利用多样路段交叉点作为锚点的兴趣点集查询方法具有良好的位置隐私保护功能;而直接查询具体路段上的 Pu_kv_n 则要暴露用户所在的路段,因此算法 2 行④同时查询 l 个随机路段上的同类兴趣点,增加用户可能分布在这些路段上的不确定性. 为了查询方便,E_i^{IS} 可以合并多个兴趣点查询请求集 $Req_{v_n}\{C_{u_i}\}$ 同时提交给 LBS 服务器,以获取每一个查询结果 Pu_iv_n,进一步增加攻击者将具体的查询请求与某个路段上的用户相关联的难度. 行⑥确保用户不变换路段的条件下,总根据用户提交的最新位置坐标返回新的 $P^{v_n}_{(K-\alpha)\mathrm{NN}}$,由于用户不断向前移动,导致经过一些兴趣点而使得 α 变小,因此 LBS 服务器可以采取增量广度搜索的方法不断扩大 $P^{v_n}_{(K-\alpha)\mathrm{NN}}$ 查询结果,在降低数据库查询操作的同时确保得到 K 个精确查询结果,行⑥～⑰为用户在某一路段的连续查询过程,直至用户离开该路段,行⑱判断如果连续查询在该有效期内未完成,则行⑲为进入新路段的该用户提供连续查询过程中的轨迹隐私保护(见算法 3),然后返回行②继续完成下一路段连续查询,直至查询有效期结束.

3.3 抗关联轨迹隐私保护方法

基于位置服务过程中,隐私主要分为 2 类:位置隐私和查询内容隐私[20-21]. 轨迹隐私(即连续位置隐私)是用户行驶位置在时间上的连续序列,某个用户连续位置隐私泄露包括 2 种方式:利用身份标识关联和利用查询请求关联. 防止身份关联可以采用假名更换方法,其中 Mix-zone 是目前普遍采用且行之有效的方法之一,然而如第 2 节相关工作所述,该方法无法抵抗查询内容关联攻击[13-14]. 本节将阐述在连续查询中如何防止攻击者利用查询内容关联和利用用户运动模式推断导致的轨迹隐私和查询隐私泄露问题.

如图 6 所示,考虑如下 2 种情况:

1) 以 v_5 为锚点的某个查询 C_i 表明用户 u_k 可能在 v_5 的某个入边上,然而在连续查询的下一时刻该查询 C_i 如果唯一利用 v_5 的下一跳顶点 v_2 来提出,即使用户进入新路段变化了假名 u_i,也表明用户一定是经过 v_5 驶向了 v_2,且此时用户在路段(v_5,v_2)上,位置隐私泄露,如此反复用户的轨迹隐私泄露. 这是由于用户利用了 v_5 作为锚点的同时暴露了其必然到达该顶点的事实,虽然用户分布在哪个路段不确定,能够保证用户的前向位置隐私,但用户到达该顶点后的后向位置隐私会由于查询内容的关联遭到推断.

2) 在某个时刻用户 u_k 离开了路段(v_9,v_5),而以 v_5 为锚点的查询 C_i 停止,而该查询以 v_2 为锚点重新出现,此时路网中用户行进情况已知,恰好 u_k 进入该路段则可以判断该查询是用户 u_k 发出的,用户位置隐私和查询隐私同时泄露. 这是由用户运动模式作为关联因素,并由查询内容关联造成的位置和查询隐私泄露,因此阻断查询内容关联是本文侧重解决的问题.

情况 1 能够确定用户所在的路段,并由连续路段暴露用户的轨迹隐私,在有一定用户分布的路径上,攻击者只知道用户经过了哪些路段,很难直接确定哪个用户发出的该查询;而情况 2 攻击者可以利用用户离开和进入路段的先后时间确定具体哪个用户发起了该查询. 针对上述 2 类问题,我们提出了抵抗连续查询内容关联和运动模式推断的轨迹隐私保护方法. 对于情况 1,要增加用户出现在 v_5 多个下一跳路口顶点的多样性(即查询 C_i 可能在 v_5 的多个下一顶点出现),需增强用户到达顶点后进入新路段的后向多样性;对于情况 2,要增加同时进入路段(v_5,v_2)用户的多样性和更换锚点后多个用户再次发起查询的时间一致性,需确保有多样查询先后进入某个路段且多个用户同时发起查询,抵抗运动模式关联带来的隐私泄露.

情况 1 的 1 种解决办法是直接向 v_5 的多个下一跳顶点注入 C_i 的同类假查询,并确保查询剩余有效期合理. 考虑到在 v_5 其他入边路段上用户可能同时存在同类查询(如一些常见查询),如果这些查询用户下一跳顶点是不同的,即可保证有其他同类查询利用不同下一跳顶点继续发起查询,从而混淆用户的轨迹隐私. 因此,可先查询 v_5 其他入边路段上是否存在同类查询能够相互混淆轨迹隐私,否则注入同类假查询.

情况 2 解决方法的整体思想是,当用户带着未完成的连续查询进入下一路段后,如果在该路段已有同类查询,则中间服务器直接在缓存中读取之前的查询结果,否则与一同进入该路段的用户构成

匿名查询组,一起向 LBS 服务器提交查询服务. 合并上述 2 种情况的中间服务器端连续查询轨迹隐私保护算法如下:

算法 3. 中间服务器端连续查询轨迹隐私保护算法.

① procedure:用户 u_k 在查询有效期内更换路段由(v_m,v_n)到达(v_n,v_w),数组 N 用来存储当前 v_n 所有入边路段上与 C_{u_k} 相同的全部查询请求用户;
② $M\leftarrow v_w$;
③ for 每个 $u_i\in N$ do
④ 调用算法 1 预估 u_i 下一跳顶点 v_i;
⑤ if $v_i\neq v_w$ then
⑥ $M\leftarrow v_i$;
⑦ end if
⑧ end for
⑨ if $|M|<\lambda_c$ then /* λ_c 为阈值 */
⑩ $M\leftarrow$随机获取 $\lambda_c-|M|$ 个 v_n 下一跳顶点;
⑪ end if
⑫ for 每个 $v_i\in M$ do
⑬ 在$Cache_{v_i}$中查找(v_n,v_i)上发出的查询请求;
⑭ if (v_n,v_i)上不存在 C_{u_k} then
⑮ 构造假查询$(u_k,loc'_{u_k},\boldsymbol{v},des,Vel,C_{u_k},T_{u_k}-T_{\text{now}})$; /* $T_{u_k}-T_{\text{now}}$为剩余查询有效期,确保路段多样性 */
⑯ 将虚假查询与其他刚进入(v_n,v_i)新用户的查询构造查询匿名组 $Group_{v_i}$;
⑰ else 在 $Cache_{v_i}$ 中读取 C_{u_k} 查询结果;
⑱ end if
⑲ end for
⑳ return $Group_{v_i}$;
㉑ end procedure

算法 3 的行③~⑪确保不少于 λ_c 个与 C_{u_k} 相同的查询用下一跳顶点发起,当每个查询 C_{u_k} 进入到新的路段后,如果缓存中有同类查询(行⑬~⑱),则直接读取查询结果,否则构造同类假查询,确保用户离开 v_n 后可能进入多个新路段. 该查询与其他刚进入的用户共同形成查询匿名组 $Group_{v_i}$,$Group_{v_i}$ 中的所有查询由中间服务器同时提交给 LBS 服务器以确保用户匿名性及查询内容多样性,不遭受情况 2 描述的关联攻击.

4 性能分析

本节主要分析路径预测、兴趣点查询和轨迹隐私保护 3 个算法在准确性、实效性、查询效率、通信量、安全性等方面的性能,并在隐私保护算法通信量小节中讨论隐私保护和查询质量之间的平衡问题.

1) 路径预测准确性

在算法 1 中,用户路径的预测采用了逐跳预测路段的方法,即中间服务器总是根据当前的行进情况、用户历史轨迹和路况预测用户下一跳到达的路段顶点,用户可在进入新路径前根据实际路况作出预测,如遇到塞车. 由于用户受路网限制,相对于自由空间更容易预知用户行进路段和位置,基于此本文将路径预测问题转化为对当前路段必达顶点的下一跳顶点的预测问题. 用户到达当前路段顶点前,由于该顶点的一跳连接顶点数是固定的,并且通常情况下该顶点连通路段不超过 4 个(即交叉路口),并且多数路口接近垂直分布,根据目的地与下一跳路段的夹角 $\theta((v_i,v_j),(v_i,des))$来选择用户可能进入的路段,这种角度差异使得预测权值的大小差异明显,同时结合了用户历史运动速度方向 $\boldsymbol{v}_m\in Vel$,这种方法通过对比预测路段与历史速度方向的一致性来提高预测准确性,因为历史轨迹总是大致指向目的地. 即使在式(3)筛选后仍有多条路段,可以继续利用式(4)选择路况更通畅、路网距离更短的路段作为预测结果,使得预测结果更符合实际情况.

2) 查询结果准确性

查询精确性是确保 LBS 服务质量的关键,算法 2 利用定理 1 和定理 2,将路段(v_m,v_n)上的用户 u_k 的 KNN 兴趣点查询转化为利用必达顶点 v_n 为查询起点的 KNN 查询集 $P^{v_n}_{(K-\alpha)\text{NN}}$和路段$(u_k,v_n)$上兴趣点集合 Pu_kv_n 的并集问题,其中集合 $P^{v_n}_{(K-\alpha)\text{NN}}$ 为假位置(锚点)的精确查询,相对于利用匿名框的方法更为精确,用户连续查询准确的关键是当前路段(u_k,v_n)兴趣点查询,由于 LBS 服务器 1 次返回给中间服务器该用户所在路段上所有该查询的兴趣点,因此中间服务器能够根据用户报告的最新位置判断用户经过了哪些兴趣点,从而根据变化的 α 继续发起 $P^{v_n}_{(K-\alpha)\text{NN}}$查询,系统将为 $P^{v_n}_{(K-\alpha)\text{NN}}$查询创建 1 个进程,使得 LBS 服务器能够按照路网距离从小到大增量返回精确 KNN 查询结果,直至查询结束.

3) 查询实效性

为 $P^{v_n}_{(K-\alpha)\mathrm{NN}}$ 创建增量查询进程并保持到查询结束是确保连续查询实效性的关键,这是由于 Pu_kv_n 被 LBS 服务器一次性返回,中间服务器很容易根据用户提交的最新位置判断经过了哪些兴趣点,因此 $\alpha=|Pu_kv_n|$ 总是逐渐变小的,而 $(K-\alpha)$ 总是逐渐变大的,即以 v_n 为查询起点的 KNN 查询集 $P^{v_n}_{(K-\alpha)\mathrm{NN}}$ 是逐渐增量变化的,需要根据 $(K-\alpha)$ 的变化让 LBS 服务器返回路网距离更远的兴趣点,因此连续查询结束前,该进程不能结束,否则影响查询实效性. 但是当 LBS 用户增加时,这会增加 LBS 服务器和中间服务器的负担,并导致实效性下降. 改进方法是一次性获取以 v_n 为查询起点的 KNN 查询集 $P^{v_n}_{\mathrm{KNN}}$,这是由于当 $\alpha=0$ 时,$(K-\alpha)$ 最大,此时用户路段前方兴趣点集为零,$Pu_k=P^{v_n}_{\mathrm{KNN}}$,即以 v_n 为起点的 KNN 查询集的上限值为 K,因此,中间服务器可以要求 LBS 服务器 1 次返回最大集并逐步发送给连续查询用户,这样可以在用户数量较多时进一步增强查询实效性和实时性. 然而,返回最大集会给 LBS 服务器带来查询负担,考虑该路段 (v_m,v_n) 上所有用户的 $P^{v_n}_{\mathrm{KNN}}$ 查询都是利用同一个固定锚点 v_n 进行,因此降低 LBS 服务器负担提高实效性的关键是中间服务器合并用户查询的能力[22]. 同时,中间服务器中设置的 $Cache_{v_n}$ 能够帮助缓存常见的查询请求,当同类查询被其他用户提交时,缓存无需提交 LBS 服务器数据库查询即可迅速返回查询结果,进一步提高查询效率.

4) 查询效率

以往利用锚点的查询方法,锚点位置是随机选取的[7-8,18-19],本文采用路网顶点作为查询锚点的方法,锚点位置总是固定的,因此每利用 1 个固定锚点获得的 1 类兴趣点查询结果可以缓存在 $Cache_{v_n}$ 中,以便其他用户在此查询时直接利用该结果,减少 LBS 服务器数据库查询操作,提高查询效率. 因此,$Cache_{v_n}$ 相关系数的设置是影响查询效率的关键,其中 $Cache_{v_n}$ 容量大小和 $Cache_{v_n}$ 中兴趣点查询结果保存时间是决定用户能否快速找到查询结果的关键因素,$Cache_{v_n}$ 中存储的常见兴趣点查询结果的比率越高,用户在 $Cache_{v_n}$ 中找到对应查询结果的概率越大. 假定 $Cache_{v_n}$ 容量 c 一定的条件下,稀有查询有 a 个,常见查询为 b 个,且 $a+b=c$, 每次到达新的查询请求中稀有查询请求的概率为 p,常见查询的概率为 q,$p+q=1$,由于 $Cache_{v_n}$ 容量固定,每增加 1 个常见查询则稀有查询相应减少 1 个,反之亦然,则常见查询充满 $Cache_{v_n}$ 的事件可以看作状态空间为 $\{0,1,2,\cdots,c\}$ 的质点随机游走问题,即质点从 a 出发到达 0 状态先于到达 c 状态的概率. 设 x_i 为质点从 i 出发到达 0 状态先于到达 c 状态的概率,则由全概率公式有:$x_i=x_{i+1}\times p+x_{i-1}\times q$,若 $u=q/p$,则可得差分方程:

$$d_x=x_i-x_{i+1}=ud_{x-1}=u^id_0, \tag{5}$$

由假设可知 $x_0=1$,$x_c=0$,则有:

$$x_0-x_c=\sum_{x=0}^{c-1}(x_i-x_{i+1})=\sum_{x=0}^{c-1}u^id_0=\frac{1-u^c}{1-u}d_0=1, \tag{6}$$

所以有:$d_0=\dfrac{1-u}{1-u^c}$,进而有:

$$x_i=x_i-x_c=\sum_{k=i}^{c-1}(x_k-x_{k+1})=\sum_{k=i}^{c-1}u^kd_0=u^i(1+u+u^2+\cdots+u^{c-i-1})d_0=\frac{u^i-u^c}{1-u}d_0=\frac{u^i-u^c}{1-r^c}, \tag{7}$$

因此 $Cache_{v_n}$ 中充满常见查询的概率 x_a 为

$$x_a=\frac{u^a-u^c}{1-u^c}=\frac{\left(\frac{q}{p}\right)^a-\left(\frac{q}{p}\right)^c}{1-\left(\frac{q}{p}\right)^c}, \tag{8}$$

由假设可知,在 $q\gg p$ 且 c 固定的条件下,a 越小 x_a 概率越高,也就是说初始状态下,如果常见查询越多,那么最终常见查询结果越能以较高的概率充满整个 $Cache_{v_n}$ 缓存空间,用户越能以较高概率在 $Cache_{v_n}$ 中快速找到兴趣点查询结果,减少更多的数据库查询操作,提高查询效率. 因此,可以采用人工提高常见查询结果在 $Cache_{v_n}$ 中的初始数量方法,如将统计数值较高的查询结果预先一次性存储在 $Cache_{v_n}$ 中,以提高查询效率. 同时,也可以采取提高常见查询在 $Cache_{v_n}$ 中保留时间 T_k 的方法,提高常见查询在 $Cache_{v_n}$ 中的保存时间,增加查询结果再次被使用的概率,从而提高查询效率. 同时,设置缓存上限、索引结构及良好的替换策略,都能够帮助提高查询效率.

另一方面,利用随机锚点查询,锚点重复利用率接近 0,因此,每次查询均需要进行 1 次新的数据库查询操作. 而利用固定锚点则能够缓存相同锚点查询,有效降低查询时间,设 1 次缓存查询耗时为 t_i,1 次数据库查询耗时为 nt_i,一段时间内用户发起查询

请求的数量为 m,其中常见查询比例为 r_c,则采用随机锚点查询耗时为 mnt_i,而用户利用缓存查询耗时为 $r_c \times mt_i + (1-r_c)mnt_i$,则利用本方法查询耗时占随机锚点查询方法耗时的百分比可表示为

$$r = \frac{r_c \times mt_i + (1-r_c)mnt_i}{mnt_i} = \frac{r_c + (1-r_c)n}{n} = \frac{r_c(1-n)+n}{n}, \quad (9)$$

由此可知,当 $n(n>1)$固定时,用户提交的查询中常见查询比例 r_c 越高,本方法查询耗时百分比 r 越低,用户查询时间越短,查询效率越高.

5) 查询数据通信量

查询通信量包括用户与中间服务器之间、中间服务器与 LBS 服务器之间的通信,其中中间服务器 1 次返回给用户精确查询结果,二者之间通信量相对固定,因此查询通信量变化总是来自中间服务器与 LBS 服务器之间的通信. 查询结果由 Pu_kv_n 和 $P^{v_n}_{(K-a)\mathrm{NN}}$ 这 2 部分组成,中间服务器出于保护用户位置隐私的需要,获取 Pu_kv_n 时不仅查询用户所在路段上的兴趣点,同时还要获取多个路段上的同类兴趣点,这会带来通信量增加.

在兴趣点均匀分布的情况下,假设等长路段上分布的某类兴趣点数量为 a,路网顶点的入边通常不超过 4 条,则每类查询的通信量增至 $4a$ 个,h 个不同查询则使通信量增至 $4ah$ 个,查询算法设置缓存 $Cache_{v_n}$ 来平衡通信量增加的缺陷,使得常用查询在被重复查询时可以在中间服务器的缓存中找到,因此利用缓存还能够在一定程度上降低通信量. 由于所有(v_m, v_n)上的用户都利用同一个锚点 v_n 查询兴趣点集,使得 LBS 服务器便于合并相关查询,因此本查询算法在考虑位置隐私保护的同时,也兼顾服务质量. 另一种降低网络通信负载的方法是在中间服务器端利用网络编码的方法,我们将在以后的研究中利用该方法.

6) 隐私保护算法的保护强度

算法 3 中连续查询用户的轨迹隐私保护主要考虑 2 种情况:①用户当前所在路段与下一路段由于查询内容关联所带来的轨迹隐私泄露;②用户进入新路段受运动模式影响先后发起查询带来的关联推断攻击,从而将轨迹与查询内容映射到具体用户上,造成更为深入的隐私泄露.

对于第 1 种情况,中间服务器会查询 v_n 顶点 M 条入边中 l 条随机入边上的兴趣点集,增加了用户 u_k 所在入边路段的不确定性,当用户到达新路段前向 v_n 顶点 L' 条出边中的 M' 条注入虚假的同类查询,则对于 M' 条出边中任意 1 条出边 i 为用户 u_k 所在路段的信息熵为

$$H(i) = -\sum_{k \in A_M} p_{i \to k} \times \mathrm{lb}\ p_{i \to k}, \quad (10)$$

其中,A_M 表示 v_n 的 M 条入边集合,$p_{i \to k}$ 表示 v_n 出边 i 上的 1 条同类查询用户 u_k 来自 v_n 入边 k 的概率. 随着连续查询在其有效期内不断被注入和当前顶点每条出边连接的多个路网顶点上,由式(1)可知,此时有多个位置序列$(loc_{\mathrm{anchor1}}, \cdots, loc_{\mathrm{anchorn}})$的权值 ε_{C_i} 较高,使得攻击者很难确定哪一组序列为用户真实轨迹,起到混淆的作用. 同时条件概率 $p_{(i \to k \mid e \to i)}$ 也呈现指数级 M^n 下降(通常 $M \leqslant 4, n \in \mathbb{N}_+$),联合信息熵升高,使得攻击者的不确定性不断升高,用户轨迹遭到查询内容关联攻击的难度不断提高. 由此,从 1 个用户离开发起查询的第 1 个顶点 v_1 开始,一直到在第 n 个顶点 v_n 的出边注入 M^n 个假查询,则任意 1 个此类查询为真实用户发起查询的信息熵 H 的理论上限为

$$H = \mathrm{lb}(|L_1 - M_1|) + n \times \sum_{2 \leqslant i \leqslant n} \mathrm{lb}\ M_i, \quad (11)$$

其中,$\mathrm{lb}(|L_1 - M_1|)$表示用户首次发起连续查询后经过第 1 个顶点 v_1 的入边信息熵,这是由于首次发起的连续查询在未更换路段前无查询关联可能,所以顶点 v_1 的所有入边具有良好的前向位置隐私安全性.

对于第 2 种情况,在新路段上合并新到查询匿名组,匿名组用户同时发起查询,由式(2)可知,存在多组 ε_{Vel} 权值较高的位置序列,攻击者推断难度增高. 假设 v_n 入边路段上分布着 A 个用户,其中 B 个用户发起连续查询,其中发起某个连续查询 Q_k 用户集合为 $U_k(u_k \in U_k)$,则对于下一路段用户构造匿名查询组中的任意用户 u_i 执行查询 Q_k 的信息熵为

$$H(u_i) = -\sum_{u_k \in U_k} p_{u_i \to u_k} \times \mathrm{lb}\ p_{u_i \to u_k}, \quad (12)$$

由此用户 u_i 位于某个路段 i 的联合信息熵:

$$H(u_i, i) = -\sum_{k \in A_M} \sum_{u_k \in U_k} p(u_k, k) \times \mathrm{lb}\ p(u_k, k), \quad (13)$$

其中,$p(u_k, k)$表示用户 u_k 位于路段 k 上的概率. 联合信息熵表明在用户注入假查询后,增加路段多样性,当用户出现在新的路段上后无论是否为假查询,均被纳入匿名查询组中并同时发起 LBS 查询,进一步增加了发起连续查询用户多样性,使得 $p(u_k, k)$ 值更小,因此联合信息熵会随用户经过的顶点增多而逐渐增大,攻击者发起关联攻击的不确定性更高.

7) 隐私保护算法的数据通信量

轨迹隐私保护算法 3 的通信量主要来自中间服务器注入假查询带来的与 LBS 服务器间的通信量增加,构造匿名查询组由中间服务器完成而没有用户与中间服务器间的沟通不会带来额外通信量.在选择注入假查询的路段数量 l 不变的条件下,注入假查询的通信量主要有 2 个因素影响:路段平均长度 S 和查询有效期 T.在查询有效期 T 不变的情况下,假查询消息总数量 Z 在查询有效期内随经过的顶点数量 n 呈现指数增长,上限为 l^n,由于注入假查询基于交叉路口数,l 通常不超过 4.一般地,在路段较长的情况下,查询有效期在少数路段甚至 1 条路段即可结束,并不会带来假查询的指数阶复制,缓存 $Cache_{v_n}$ 也会降低注入假查询带来的查询通信量.简化路网图中不必要的顶点,去掉一些连通路段少的路口顶点,可以使查询路段变长,减少经过顶点过多带来的假查询不断被复制的数量;但这会使得分布在这些路段上的兴趣点被查询忽略,影响查询质量.如果用户以平均速度 $\boldsymbol{v}$ 行驶,在平均路段长度 S 一定的条件下,查询有效期 T 越长,假查询数据量 Z 增长越快:

$$Z = l^{\lfloor T \bmod (S/|\boldsymbol{v}|) \rfloor}. \tag{14}$$

本方法面对查询有效期较长、平均路段较短的连续查询时,需要平衡轨迹隐私保护质量和由此带来的查询负担.假设在每个顶点注入假查询数 l 和用户平均速度 $\boldsymbol{v}$ 保持不变,当查询有效期 T 超过某个阈值 $T_{\max}$ 时,中间服务器须要去掉一些不必要顶点,使得路网中路段长度 S 变大,限制 Z 在一定范围 $Z_{\max}$ 内增长,若在单位长度为 S 的路段上某类兴趣点数量为 g,中间服务器去掉不必要的每个顶点最多有 2 条出边路段被忽略,则将 $T_{\max}$,$Z_{\max}$ 带入式(14)求出 $S'=S$ 后,对应带来的兴趣点丢失数量下限为 $2g(\lfloor S'/S \rfloor - 1)$.因此,由于去掉部分路口顶点延长 S 带来的查询兴趣点丢失比率 r_i 为

$$r_i = \frac{2g(\lfloor S'/S \rfloor - 1)}{g[3(\lfloor S'/S \rfloor + 1) + 1]} = \frac{2(\lfloor S'/S \rfloor - 1)}{3(\lfloor S'/S \rfloor + 1) + 1}, \tag{15}$$

兴趣点丢失率与 g 无关,S' 越大带来的丢失比率越高,当 S' 趋于无穷大时,丢失率为 66.67%.因此,在遇到较大有效期的连续查询请求时,算法 3 总能够保证注入假查询总数量 Z 不超过一定范围,代价是查询精确率的下降.

8) 隐私保护算法的实时性

影响算法 3 的实时性因素主要来自于构造匿名查询组带来的时间延迟.由于先进入新路段的用户需要等待匿名组所有其他用户都进入路段后一起发起查询,这个时间差 Δt 是影响先进入用户查询效率的主要因素.中间服务器可以利用用户最新更新的位置情况为用户预先构造匿名组,并要求用户在将要进入新路段的时间段内高频率报告最新的位置直至进入新路段.因此,不同用户更新位置的频率差异不会影响实时性.而匿名查询组的第 1 个用户与最后 1 个用户进入新路段的时间差 Δt 为二者进入路段前路网距离 Δs 与最后进入用户速度 $|\boldsymbol{v}|$ 之比,通常中间服务器构造匿名组会选择位置相邻用户,在用户非极端稀疏的情况下,Δt 相对较小.

5 实　　验

本节主要关注用户在获取 LBS 过程中,受用户数量、查询兴趣点数量、查询有效期和兴趣点集保留时长等多因素影响下,通信量等指标的变化情况.实验在模拟数据集上进行,首先说明在上述多因素变化情况下数据通信量及服务响应时间的变化情况,然后通过与经典位置隐私保护方法对比说明本方法的优势.

5.1 实验环境配置

实验在 Windows 7 系统上用 Java 语言实现,运行硬件环境为 3.2 GHz Intel Core i5 处理器、4 GB 内存.实验使用 Thomas Brinkhoff 路网数据生成器[①]生成连续查询中的每次快照查询,实验共生成 20 万条路网环境下用户轨迹,用户随机均匀分布在每条路网上,每个用户的出发地和目的地是随机的,速度受路段限制影响.路网地图来自美国国家地理勘探局(USGS)[②]提供的亚特兰大市路网,实验过程中取该市及附近地区的 9 286 条路段和 6 927 个路段交叉点,被选作锚点数的交叉点默认值约为总交叉点数的 50%.实验的网络通信带宽为 3 MBps,每次返回单个数据包为 256 B,除去其中 40 B 的包头,每次返回的数据包中包含兴趣点个数约为(256−40)/8=27 个.中间服务器将兴趣点查询结果保存

① http://iapg.jade-hs.de/personen/brinkhoff/generator

② http://www.usgs.gov

在存储空间 $Cache_{v_n}$ 中，$Cache_{v_n}$ 中的数据集由 2 KB 的 R-tree 结构索引，参数配置情况如表 1 所示：

Table 1 Default Configuration Parameters

表 1 实验默认参数配置

Parameter Name	Value Scope	Default Value
Number of Users U	[50 000, 300 000]	100 000
Query Number of POIs K	[10, 60]	20
Average Query Validity Period T_L/min	[0, 20]	5
User's Location Updating Frequency f /min	[0.25, 2]	0.5
Retention Time of POIs in $Cache_{v_n}$ T_k/h	[1, 12]	6
Ratio of Common Query in $Cache_{v_n}$ r_c/%	[30, 90]	75
Number of Fake Query Injection z	[1, 4]	2
Average Length of Road Segments S /m	[200, 2 000]	1 000

5.2 本方法实验性能

由表 1 可以看出，数据通信量的影响因素是多方面的，由第 4 节性能分析可知，由于中间服务器代理用户发送查询，用户与中间服务器除位置周期更新外的通信量较少，所以通信量主要来自中间服务器与 LBS 服务器间的通信，在不同参数影响下中间服务器向 LBS 服务器提出的查询请求量不同，表现出较大差异的通信量.

如图 7 所示，当用户数量递增时，平均数据包通信量受兴趣点在中间服务器 $Cache_{v_n}$ 中保留时间 T_k 的影响，平均通信量总体表现为随用户数量增多而增长. 当 $T_k=[1,3]$时，数据包通信数量接近指数增长，这是中间服务器注入假查询而带来的中间服务器与 LBS 服务器间通信造成的，在默认查询有效期 T_L 和假消息数量 z 不变的条件下，增长表现较为明显；而当 T_k 逐渐增大时，意味着有更多的兴趣点集合能够在中间服务器的缓存 $Cache_{v_n}$ 中找到，增长方式逐渐变化为多项式级；在 $T_k=[9,12]$时随着用户数量增大甚至出现了通信消息数量下降的趋势，这是由于更多用户带来了多样查询结果使得 $Cache_{v_n}$ 中保存的兴趣点类型增多，甚至对于一些稀有的兴趣点请求也能够长时间保留下来，减少了 LBS 服务器与中间服务器间的通信量. 因此，增大 $Cache_{v_n}$ 容量及兴趣点结果集保存时间，能够有效降低由于注入假查询带来的额外通信；而分布式保存兴趣点查询结果集及其有效的索引结构能够降低中间服务器兴趣点查询时延.

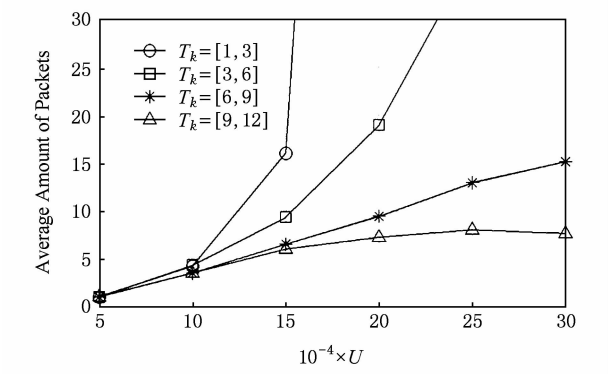


Fig. 7 U and T_k variation.

图 7 U 及 T_k 变化

图 8 显示用户的位置更新频率变化不会带来通信量的明显变化，这是由于查询算法首先查找当前路段用户前方所有兴趣点 Pu_kv_n，并以此要求 LBS 服务器 1 次返回剩下的以 v_n 为锚点的全部兴趣点集 $P^{v_n}_{(K-\alpha)\mathrm{NN}}$. 因此，用户频繁更换位置，中间服务器只需根据其位置计算变化的$(K-\alpha)$NN 个兴趣点返回给用户即可，不会带来频繁的 LBS 服务器端数据库查询操作请求，用户行驶到新路段时才会进行 1 次该操作.

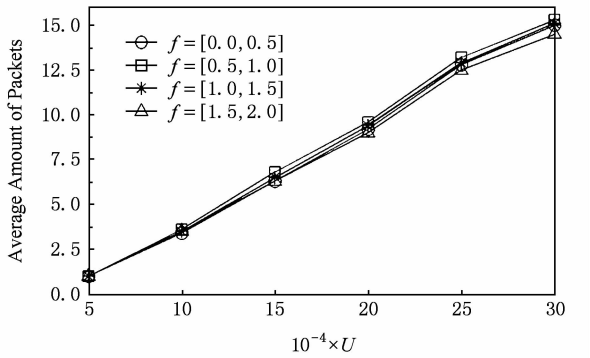


Fig. 8 U and f variation.

图 8 U 及 f 变化

另一方面，当 K 发生变化时，中间服务器与 LBS 服务器通信更加频繁，如前所述每个从 LBS 服务器返回的数据包最多可以包含 27 个兴趣点信息，

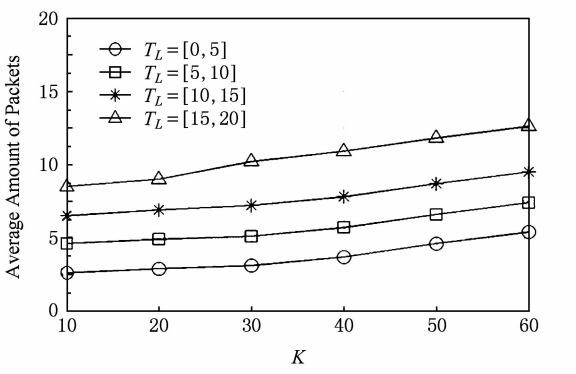


Fig. 9 K and T_L variation.

图 9 K 及 T_L 变化

当 K 逐渐增大时,需要返回更多数据包来容纳查询到的兴趣点,因此通信量总体会上升. 如图 9 所示,当平均查询有效期 T_L 不断增长的过程中,消息数量会随之逐渐增加,由于默认 $Cache_{v_n}$ 中保留的兴趣点时间最长为 6 h,消息数量并没有随 T_L 增长发生特别明显的增长,这是由于去掉了部分次级路段交叉点的原因,但会带来一定的兴趣点查询精度下降,兴趣点丢失度见图 10.

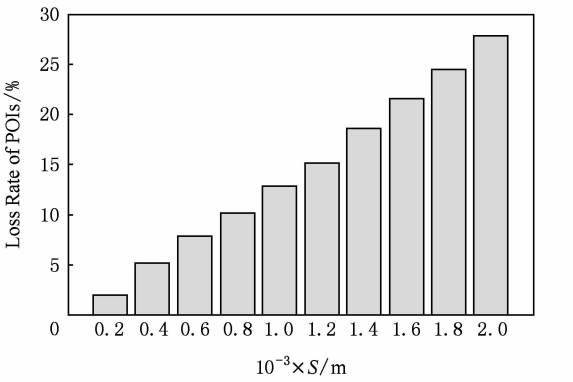


Fig. 10 Loss rate of POIs result.

图 10 兴趣点查询丢失率

由式(14)可知,当 T_L 不断增加时,路段长度 S 需调整以使用户选择注入假查询路段数下降,提高查询效率,但会带来撤销路段兴趣点丢失. 当 $200\leqslant S\leqslant 2\,000$ 时,丢失率不超过 30%;而当路段长度 S 无限增长时,兴趣点丢失率升至 66.7%. 中间服务器注入假查询消息数量 z 对通信量产生影响,如图 11 所示:

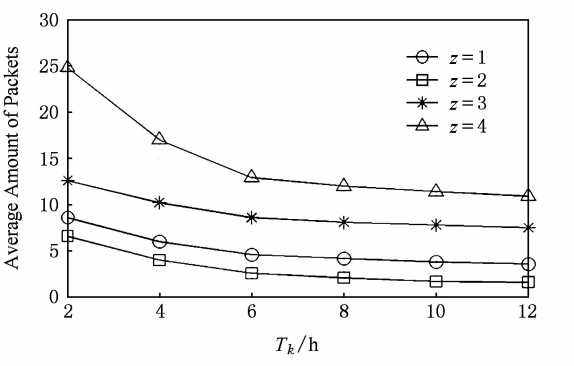


Fig. 11 T_k and z variation.

图 11 T_k 及 z 变化

由式(5)~(9)可知,缓存 $Cache_{v_n}$ 中常见查询的比例 r_c 越高,用户查询效率越高,用户越容易快速找到查询结果,降低数据包发送量. 如图 12 所示,用户数量逐渐增多时,平均数据包量随之增加;而 r_c 比例越高,平均数据包量越低,与性能分析结论相符.

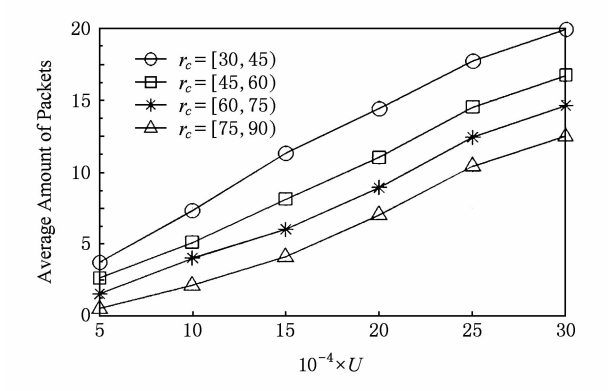


Fig. 12 U and r_c variation.

图 12 U 及 r_c 变化

综上,本方法在查询及隐私保护 2 个主要过程中,由于受 T_L,T_k,r_c 等多因素的影响,平均数据包量不断变化,有效平衡多个因素间的关系,可以使查询和位置隐私保护质量得到保障的同时有效控制通信量不过度增长.

5.3 对比实验

本节选取 2 个利用锚点查询来保护用户位置隐私的经典方法 SpaceTwist 和 KAWCR[18] 与本文提出的方法比较,通过对比每次查询的平均数据包发送量、平均查询响应时间、查询准确率及算法隐私保护强度等诸多影响因素,说明本方法具有的优势.

如图 13 所示,当用户数量增加时,3 种方法的平均数据包量均有增加;在其他默认设置相同条件下,本文方法通信量增加并不明显,主要原因是本文方法采用路口交叉点作为锚点,该锚点位置是固定的,这便于中间服务器和 LBS 服务器分别合并同类查询,同时固定锚点也为同一条路段上兴趣点结果再次被利用带来方便,而其他 2 种方法采用随机选取锚点的方式. 当用户数量增加时,不同的锚点会带

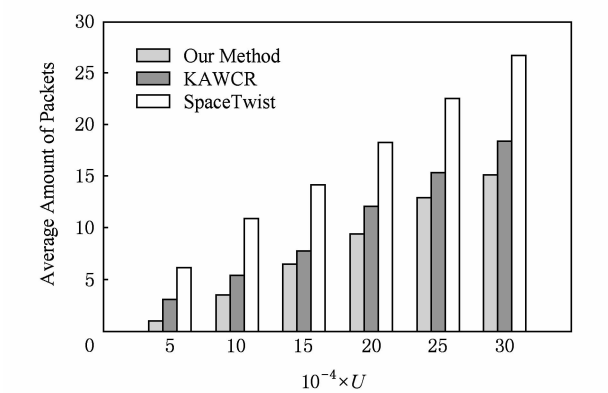


Fig. 13 Average packets trend with U variation.

图 13 平均消息数量随 U 的变化

来频繁的 LBS 服务器数据库查询操作,相应的平均响应时间变化如图 14 所示:

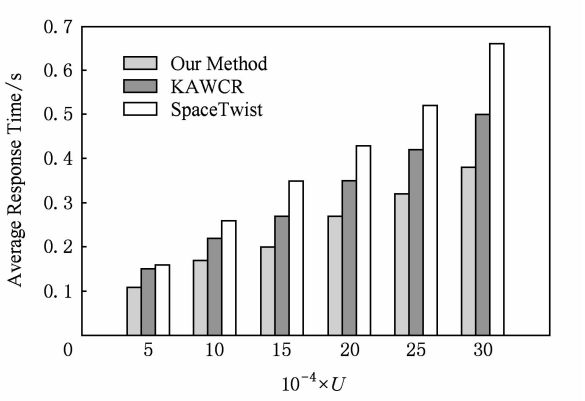


Fig. 14 Average response time trend with U variation.

图 14 平均响应时间随 U 的变化

图 15 显示在用户查询兴趣点数量 K 递增时,本文方法数据包量变化不明显,而 SpaceTwist 方法数据包量增长明显.这是因为 SpaceTwist 方法不断扩大查找范围,较多距离用户较远的兴趣点被查出,带来通信量增长;而本文方法采用固定锚点查询,在路网图中采用广度优先搜索遍历的方式直接获取精确的 KNN 兴趣点集.

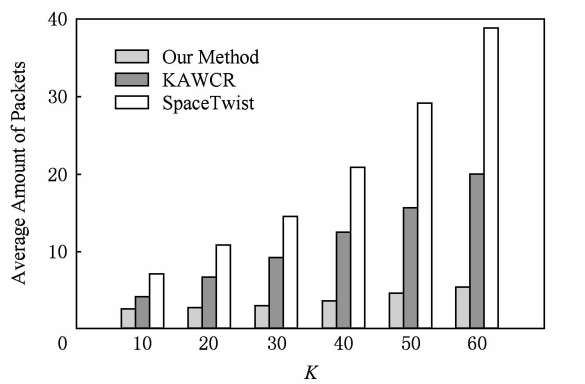


Fig. 15 Average packets trend with K variation.

图 15 平均消息数量随 K 的变化

如图 16 所示,用户 u_k 与锚点距离 $dist(u_k,q)$ 发生变化时,SpaceTwist 方法数据包量上升较快;本文方法与 KAWCR 方法具有较低近似平均消息数量,且相对较为稳定.这是由于本文方法在查询过程中不受用户与锚点距离的影响,LBS 服务器以路口顶点为锚点返回兴趣点集是精确的,无额外查询导致的消息数量增加.

查询准确率是指经保护算法处理的 n 次兴趣点查询结果中准确结果的比率.如图 17 可以看出,当用户数量变化时,本文方法的查询准确率总是接近 100%,这是由于本文方法采用路网顶点作为固定锚点,广度遍历算法总能以该锚点作为出发点,获取精确的兴趣点 K 近邻查询结果;而由于 SpaceTwist 方法在查询过程中存在查找不均衡问题[23],因此查询准确率较低,且该方法采用随机选取锚点的方法,因此在用户数量增长的过程中,查询结果的准确率大大降低;KAWCR 方法的准确率介于二者之间.

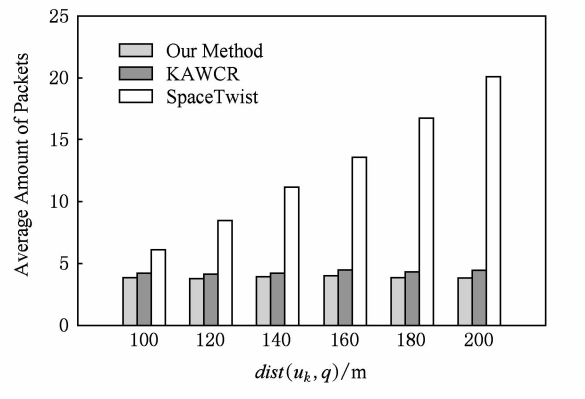


Fig. 16 Average packets trend with $dist(u_k,q)$ variation.

图 16 平均消息数量随 $dist(u_k,q)$ 的变化

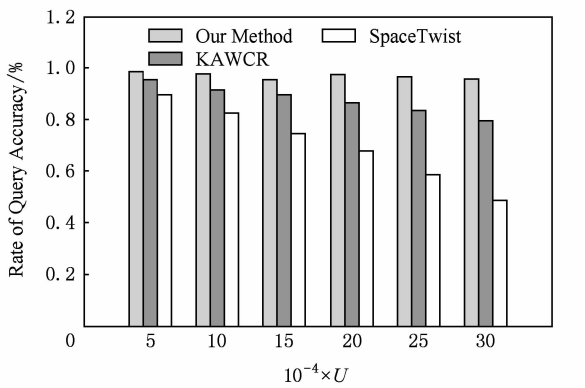


Fig. 17 Query accuracy rate with U variation.

图 17 查询准确率随 U 的变化

由式(10)(11)可知,当顶点 v_n 有 4 条入边时,处在顶点 v_n 的某条出边上的用户,由其查询内容关联推断出用户来自于该顶点 v_n 的某条入边的信息量为 2 b,当用户连续经过 5 个顶点后(通常情况下,用户 1 次查询有效期内经过的路口顶点数不超过 5 个)信息量至少为 10 b,以此时信息量值为界限,如果用户在 1 次查询有效期内连续查询信息量低于 10 b,则可以视为存在被关联推断可能.高于 10 b 的查询占全部查询的请求比率如图 18 所示.

由图 18 可以看出,当用户数量不断增长时,3 种方法的信息量比率均表现出了增长,这是由于用户数量增加增大了系统的复杂度和不确定性,该比率

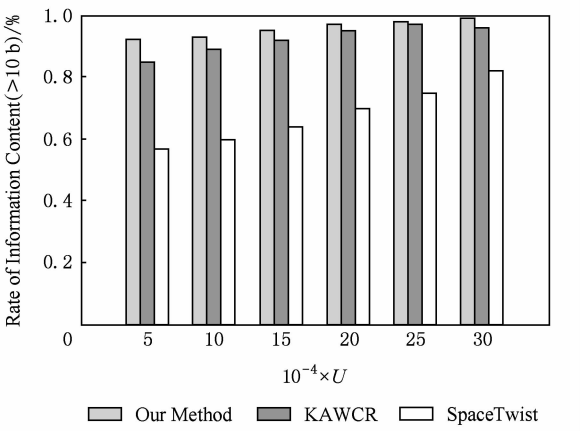


Fig. 18 Ratio of information content with U variation.

图 18 信息量大于 10 b 的查询请求比率随 U 的变化

越高,说明有更多的查询具有较强的抗内容关联性,算法的隐私保护强度越好;而 SpaceTwist 的初始比率值较低,这是由于该算法没有充分考虑查询内容关联可能带来的关联攻击问题.同理,基于用户运动模式的推断问题可以由式(12)(13)得出类似结论,这里不做赘述.

6 结 论

本文研究了路网环境下能够保护用户位置隐私的兴趣点查询方法,该方法以路网交叉点为锚点代替用户真实位置查询,将用户 KNN 兴趣点查询转化为等价的以锚点为出发原点的 K 个兴趣点查询问题.然后,基于注入假查询和构造查询匿名组的方法,提出了抗查询请求关联攻击和抗运动模式推断攻击的轨迹隐私保护方法,并分析了在查询有效期较长时如何平衡隐私保护度和查询准确性的问题.实验和性能分析证明了本方法具有较好的工作效率和相对优势.

参 考 文 献

[1] Gruteser M, Grunwald D. Anonymous usage of location-based services through spatial and temporal cloaking [C] //Proc of the 1st Int Conf on Mobile Systems, Applications and Services. New York: ACM, 2003: 31-42

[2] Chow C Y, Mokbel M F. Trajectory privacy in location-based services and data publication [J]. ACM SIGKDD Explorations Newsletter, 2011, 13(1): 19-29

[3] Chow C Y, Mokbel M F. Enabling private continuous queries for revealed user locations [G] //Advances in Spatial and Temporal Databases. Berlin: Springer, 2007: 258-275

[4] Kim H I, Shin Y, Chang J W. A grid-based cloaking scheme for continuous queries in distributed systems [C] //Proc of the 11th IEEE Int Conf on Computer and Information Technology (CIT). Los Alamitos, CA: IEEE Computer Society, 2011: 75-82

[5] Wang Y, Peng J, He L, et al. LBSs privacy preserving for continuous query based on semi-honest third parties [C] //Proc of the 31st IEEE Int Conf on Performance Computing and Communications (IPCCC). Los Alamitos, CA: IEEE Computer Society, 2012: 384-391

[6] Bamba B, Liu L, Pesti P, et al. Supporting anonymous location queries in mobile environments with privacygrid [C] //Proc of the 17th Int Conf on World Wide Web. New York: ACM, 2008: 237-246

[7] Kido H, Yanagisawa Y, Satoh T. An anonymous communication technique using dummies for location-based services [C] //Proc of the 2005 Int Conf on Pervasive Services. Los Alamitos, CA: IEEE Computer Society, 2005: 88-97

[8] Yiu M L, Jensen C S, Huang X, et al. SpaceTwist: Managing the trade-offs among location privacy, query performance, and query accuracy in mobile services [C] //Proc of the 24th IEEE Int Conf on Data Engineering (ICDE 2008). Los Alamitos, CA: IEEE Computer Society, 2008: 366-375

[9] Dewri R, Ray I, Whitley D. Query m-invariance: Preventing query disclosures in continuous location-based services [C] //Proc of the 11th Int Conf on Mobile Data Management (MDM 2010). Los Alamitos, CA: IEEE Computer Society, 2010: 95-104

[10] Pan X, Meng X, Xu J. Distortion-based anonymity for continuous queries in location-based mobile services [C] //Proc of the 17th ACM SIGSPATIAL Int Conf on Advances in Geographic Information Systems. New York: ACM, 2009: 256-265

[11] Pan X, Xu J, Meng X. Protecting location privacy against location-dependent attacks in mobile services [J]. IEEE Trans on Knowledge and Data Engineering, 2012, 24(8): 1506-1519

[12] Beresford A R, Stajano F. Mix zones: User privacy in location-aware services [C] //Proc of the 2nd IEEE Annual Conf on Pervasive Computing and Communications, Workshops (PerCom 2004). Los Alamitos, CA: IEEE Computer Society, 2004: 127-131

[13] Palanisamy B, Liu L, Lee K, et al. Anonymizing continuous queries with delay-tolerant mix-zones over road networks [J]. Distributed and Parallel Databases, 2014, 32(1): 91-118

[14] Pingley A, Zhang N, Fu X, et al. Protection of query privacy for continuous location based services [C] //Proc of the 2011 Int Conf on Computer Communications. New York: IEEE Communications Society, 2011: 1710-1718

[15] 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 Databases. New York: ACM, 2006: 763-774

[16] Hashem T, Kulik L, Zhang R. Countering overlapping rectangle privacy attack for moving *k*NN queries [J]. Information Systems, 2013, 38(3): 430-453

[17] Lin Xin, Li Shanping, Yang Zhaohui. Attacking algorithms against continuous queries in LBS and anonymity measurement [J]. Journal of Software, 2009, 20(4): 1058-1068 (in Chinese)
(林欣, 李善平, 杨朝晖. LBS 中连续查询攻击算法及匿名性度量[J]. 软件学报, 2009, 20(4): 1058-1068)

[18] Gong Z, Sun G Z, Xie X. Protecting privacy in location-based services using *k*-anonymity without cloaked region [C] //Proc of the 11th Int Conf on Mobile Data Management (MDM 2010). Los Alamitos, CA: IEEE Computer Society, 2010: 366-371

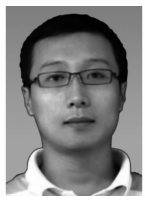
[19] Hang Yi, Huo Zheng, Meng Xiaofeng. CoPrivacy: A collaborative location privacy-preserving method without cloaking region [J]. Chinese Journal of Computers, 2011, 34 (10): 1976-1985 (in Chinese)
(黄毅, 霍峥, 孟小峰. CoPrivacy: 一种用户协作无匿名区域的位置隐私保护方法[J]. 计算机学报, 2011, 34(10): 1976-1985)

[20] Ghinita G. Privacy for location-based services [J]. Synthesis Lectures on Information Security, Privacy, & Trust, 2013, 4(1): 1-85

[21] Pan Xiao, Hao Xing, Meng Xiaofeng. Privacy preserving towards continuous query in location-based services [J]. Journal of Computer Research and Development, 2010, 47 (1): 121-129 (in Chinese)
(潘晓, 郝兴, 孟小峰. 基于位置服务中的连续查询隐私保护研究 [J]. 计算机研究与发展, 2010, 47(1): 121-129)

[22] Elmongui H G, Mokbel M F, Aref W G. Continuous aggregate nearest neighbor queries [J]. GeoInformatica, 2013, 17(1): 63-95

[23] Zhou C, Ma C, Yang S, et al. A location privacy preserving method based on sensitive diversity for LBS [C] //Proc of the 11th IFIP Int Conf on Network and Parallel Computing. Berlin: Springer, 2014: 409-422



Zhou Changli, born in 1985. Received his PhD degree from Harbin Engineering University. Member of China Computer Federation. His main research interests include privacy preserving in LBS and information security(zhouchangli666@163.com).

Ma Chunguang, born in 1974. Received his PhD degree from Beijing University of Posts and Telecommunications. Senior member of China Computer Federation. His main research interests include cryptography, information security and privacy preserving.

Yang Songtao, born in 1972. Associate professor and PhD candidate. Member of China Computer Federation. His main research interests include privacy preserving in LBS and information security(songtao_y@163.com).