DOI:10.7631/issn.1000 - 2243.2013.06.0975 **文章编号**:1000 - 2243(2013)06 - 0975 - 06

基于个性化隐私需求的查询隐私保护算法研究

孙 岚, 周 浩, 吴英杰, 王一蕾

(福州大学数学与计算机科学学院, 福建 福州 350116)

摘要: 现有大多数基于位置服务(location based service,LBS)的隐私保护算法都将对用户位置隐私的保护等同于对整个 LBS 查询服务隐私的保护. 但是, 在用户位置信息已知的前提下, 这些算法有可能面临推断攻击. 在考虑用户个性化隐私需求的情况下, 基于四分树结构提出了能够避免此类推断攻击的隐私保护算法; 为了有效的减小隐惹区域的大小基于半象限的定义对该算法进行了进一步优化. 最后, 通过仿真实验验证了算法抵御推理攻击的有效性.

关键词: 查询隐私保护; 个性化隐私需求; 隐匿区域; 四分树; 半象限

中图分类号: TP311 **文献标识码**: A

Research of query privacy protection algorithm with personalized requirements of privacy

SUN Lan, ZHOU Hao, WU Ying - jie, WANG Yi - lei

(College of Mathematics and Computer Science, Fuzhou University, Fuzhou, Fujian 350116, China)

Abstract: Location privacy protection and query privacy protecton is considered to be equivalent in most existing LBS(location - based service) privacy protection algorithms. However, under the premise of the user location information is known, these algorithms will lead to inferring attack. In this paper, we proposed an algorithm based on quadtree to avoid this kind of attack, and to satisfy the user's personalized query privacy. Furthermore, in order to rerude the size of cloaking area, we optimized the algorithm based on half quadrant defination. Finally, the effectiveness of the algorithms to resist inferring attack is verified by simulation experiment.

Keywords: query privacy protection; personalized requirements of privacy; cloaking area; quadtree; half quadrant

0 引言

伴随着无线通讯技术和网络技术的迅猛发展, 一种基于位置的服务(location based service, LBS)悄然而生. 近年来, 关于 LBS 的应用随处可见, 它给人们的生活带来了不容忽视的便利. 然而, 在人们享受服务的同时, 也不得不面对个人隐私可能被泄漏的危险. 因此, 在为用户提供基于位置服务的同时, 考虑如何保护用户的个人隐私安全成为近年来数据安全领域的研究热点之一[1].

基于位置的服务需要用户提供精确的位置信息和查询内容, 因此有关 LBS 的个人隐私安全通常包括两个方面, 一是用户位置的隐私安全[2], 二是用户所查询内容的隐私安全. 位置隐私安全是指发出服务请求的用户所在的精确位置应该受到保护, 从这方面来看, 位置的泄漏等价于用户个人隐私的泄漏; 当用户的实际位置作为公开信息提供时, 用户发出的查询内容也可能成为泄漏个人隐私的一个重要原因, 所以需要对查询内容进行隐私保护[3], 这样即便攻击者获得了用户所在的具体位置, 也无法获知该用户发出了怎样的请求. 目前关于 LBS 的隐私保护研究大多集中于对位置隐私的保护, 究其原因, 大多数的研究者认为对位置隐私的保护等同于对整个查询服务的保护, 即不区分位置隐私和查询隐私[4-10]. 然而在一些实际应用中, 用户的位置信息是公开的, 此时隐私保护的重点是用户基于位置发出的查询, 在这种情况下, 若将之前大多数的研究成果直接应用于此类问题, 都会使用户的查询隐私受到威胁[3]. 如何在用户位

收稿日期: 2013 - 04 - 08

通讯作者: 孙岚(1978 -), 讲师, E - mail: lsun@fzu.edu.cn

基金项目: 福建省自然科学基金资助项目(2010J01330); 福州大学科技发展基金资助项目(2012 - XQ - 27)

置信息公开的前提下，考虑不同用户个性化的隐私需求，设计有效的隐私保护算法来保护用户的查询隐私安全成为一个至关重要的问题.

1 相关工作

现有关于 LBS 隐私保护问题的研究大多采用位置 k - 匿名模型[4]，即将一个用户 q 的位置信息与其它 $k-1$ 个用户的位置信息构成一个匿名集. 此类研究中构造包含用户的匿名集常采用空间隐匿(spatial cloaking)的方法，即用一个包含查询用户的封闭区域(一般为矩形)取代该用户，封闭区域中包括除了查询用户以外的至少 $k-1$ 个其他用户. 研究的重点多集中于如何设计有效的空间隐匿算法.

文献[4]提出了一种基于四分树(quad - tree)结构的 Interval - Cloak 算法，该算法将整个空间区域自顶向下递归地划分成四个等象限的形式，直到用户所在象限中包含的用户数少于系统设定的 k，然后把该象限的父象限作为查询的空间隐匿区域. 这种方法的缺点是产生的隐匿区域容易过大，从而降低了服务质量，且不支持用户的个性化隐私需求. 文献[5]提出了一种基于图论的 Clique - Cloak 算法，该算法考虑了用户的个性化隐私需求和服务质量需求，允许用户定义自己的匿名度 k 和最大模糊化区域 Amax. 但该算法只适用于 k 值很小的情况，当 k 值越大时，匿名成功率越低. 文献[6]针对 Interval - Cloaking 算法产生的隐匿区域容易变得过大的缺点，提出了分别基于完全金字塔和不完全金字塔数据结构的基本 Casper 算法及自适应 Casper 算法，且支持用户的个性化隐私需求. 算法使用自底向上的方法寻找空间隐匿区域. 当用户所在的象限不满足用户的隐私需求 k 时，自适应的 Casper 算法首先考虑结合它的相邻象限，如果通过结合它的相邻象限仍不能满足隐私需求 k，再考虑它的父象限，直到满足用户的隐私需求为止. 显然，自适应 casper 算法相对于 Interval - Cloak 算法而言，有效的减小了隐匿区域的可能大小. 上述算法都只适合于均匀分布的数据，如果用户的位置分布是非均匀的，则存在异常点攻击，可能导致用户查询隐私泄露. 因此，文献[7]提出了一个附加的条件 k - reciprocity 防止这种异常点攻击. k - reciprocity 是指当一个用户发送请求时，不仅要求产生的空间隐匿区域要至少包含其他 $k-1$ 个用户，而且要求在此空间隐匿区域中的所有请求用户都要把该空间区域作为他们的共同隐匿区域.

上述文献中的算法都是把对位置隐私的保护等同于对整个 LBS 查询服务的保护，但是由于受到现实情况的影响，上述算法并不能够有效保证用户的查询隐私安全[3].

文献[3]对位置隐私和查询隐私进行了严格区分，提出了一个附加的约束条件 k - sharing 防止异常点攻击. k - sharing 是指在同一空间隐匿区域内的请求用户中至少要有 k 个请求用户把此空间区域作为他们的共同隐匿区域，而其他的请求用户则可以使用其它的空间区域作为他们的隐匿区域. 但是文献[10]指出即使满足附加的约束条件 k - reciprocity 或 k - sharing 也不能防止基于攻击者知道匿名策略的攻击. 为了防止这样的攻击者，文献[10]提出了一种能够获得最优的匿名策略的多项式算法.

2 相关定义

定义 1 位置 k - 匿名是指一个用户的位置无法与其他 $k-1$ 个用户的位置相区别[4].

位置 k - 匿名的基本思想是：用包含用户本身及其它 $k-1$ 个用户在内的隐匿区域替代用户的精确位置，使得用户的位置信息模糊于多个用户之中. 攻击者识别真实用户位置信息的概率降至 $1/k$，从而保证了移动用户的位置隐私安全.

为了满足位置 k - 匿名，通常采用的方法是形成一个关于用户 q 的隐匿区域，该区域中包括至少 k 个不同的用户. 如图 1 所示，用户 A 的精确位置由矩形隐匿区域取代，在该区域中包括 A、B、C、D 四个用户，因此满足位置 4 - 匿名.

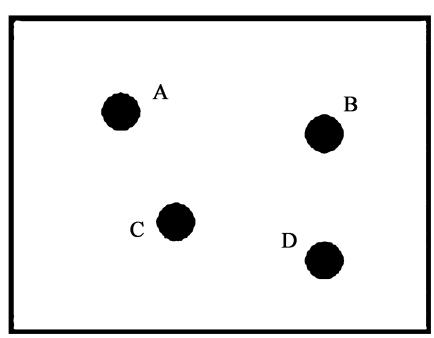


图 1 空间隐匿区域

Fig. 1 Spatial cloaking area

定义 2 为了明确用户个性化的查询请求，利用五元组来表示每一个具体的查询请求 Q.

$$Q = < \text{uid}, l, k, r, t >$$

其中：uid 表示用户的标识信息，通常是用户的姓名；$l=(x, y)$ 表示用户的坐标信息，反映了用户的具体位置；k 即用户提出的隐私要求，该信息满足了用户隐私要求的个性化；r 为一个具体的查询内容，如“离我最近的银行”；t 表示用户发送查询请求的时间.

定义 3　为了实现对用户提出的查询请求进行匿名保护，将用户基于某一确定位置提出的查询请求隐匿在一个区域中，用一个四元组来形式化地表示这一匿名请求 AR.

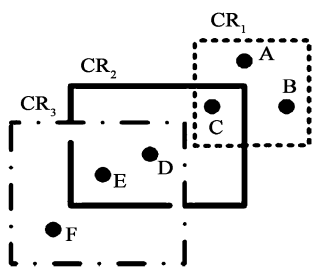
$$AR = < rid, CR, r, t >$$

其中：rid 表示发出请求的用户 id；CR 代表隐匿区域，通常选择矩形，此时，CR = <(x1, y1), (x2, y2)>，(x1, y1)是矩形区域的左下角位置坐标，(x2, y2)是矩形区域的右上角位置坐标；r，t 同查询请求 Q 中的含义，分别代表查询内容和发送查询请求的时间.

定义 4　假设 AR 是一个查询请求 Q 的匿名请求，如果匿名请求的空间隐匿区域中至少包含了 k 个用户，且使得攻击者无法以大于 $1/k$ 的概率确认是由谁发送的查询请求，那么称此匿名请求满足请求 k－匿名. 满足位置 k－匿名并不一定也满足请求 k－匿名.

3　推断攻击

攻击者可以通过三角测量、公开信息、物理观测等方法获取被攻击者附近的用户位置；而且在服务提供商不可靠的前提下，查询位置可能被泄漏. 如图 2 所示，有 A、B、C、D、E、F 六个不同的查询用户. 在位置 3－匿名模型下，用户 A、B、C 以 CR_1 作为他们的空间隐匿区域，用户 D、E 以 CR_2 作为其空间隐匿区域，用户 F 以 CR_3 作为他的空间隐匿区域. 在攻击者知道用户的位置分布情况下，攻击者可以推测出从 CR_3 中提出的查询，一定是由用户 F 提出的. 因为如果是 CR_3 中的其他用户 D 或 E 提出的查询请求，则生成的空间隐匿区域应该是 CR_2，而不是 CR_3. 用户 F 的查询隐私受到攻击.



用户	隐匿区域
A, B, C	CR_1
D,E	CR_2
F	CR_3

图 2　查询隐私泄漏

Fig. 2　The exposure of query privacy

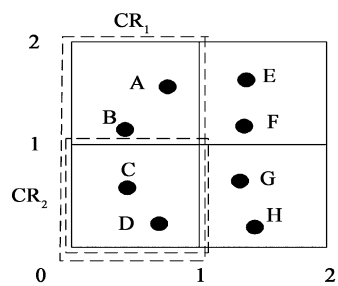


图 3　推断攻击

Fig. 3　Inferring attack

如图 3 所示，已知有 A、B、C、D、E、F、G、H 共 8 个用户，在同一时刻 t，用户 A 提出了 $k=4$ 的查询请求 Q_1，用户 C 提出了 $k=2$ 的查询请求 Q_2. 假设利用 Casper 算法分别得到了 CR_1 = <(0, 0), (1, 2)> 和 CR_2 = <(0, 0), (1, 1)> 作为 Q_1 和 Q_2 的隐匿区域. 从位置 k－匿名的角度可知，攻击者识别用户 A 的概率大于等于 1/4. 但是，位于隐匿区域 CR_2 中的用户 C(或 D)不可能同时发出两个不同的查询请求 Q_1 和 Q_2，假设攻击者已知用户的位置分布情况，那么攻击者可以判断出在 t 时刻，CR_2 中至多只有一人可能发出查询请求 Q_1，从而使得 CR_1 中发出查询 Q_1 的用户数至多为 3 人，即 A、B 和 C(或 D)，攻击者识别用户 A 查询请求的概率大于等于 1/3，即无法满足请求 4－匿名. 因此，在极端情况下，攻击者有可能以 100% 的概率推断出在 t 时刻发出请求的具体用户，用户的查询内容受到了推断攻击.

4　防止推断攻击的隐私保护算法

4.1　算法思想

推断攻击的存在意味着在满足位置 k－匿名的同时无法满足请求 k－匿名. 存在推断攻击的主要原因在于：在同一时刻，为不同查询请求 Q_i 和 Q_j 形成的隐匿区域 CR_i 包含 CR_j，且 CR_i 中的用户数正好为 k 个. 如果在图 3 所示的问题中，为查询 Q_1 形成的隐匿区域是 CR_3 = <(0, 1), (2, 2)>，那么攻击者无法

确定发出请求的用户是 A、B、E、F 中的哪一个，即满足了请求 4 - 匿名，上述的推断攻击不复存在.

考虑用四分树的根节点表示整个研究区域，每个非叶节点都有四个儿子节点，且四个儿子节点对应的区域是将其父节点对应的区域进行四等分得到的四个不同象限. 每个叶节点表示划分的最小区域，若用 cnode 来表示某个节点，那么每个节点上的用户数为 cnode. Num. 对任意用户 u 发出的查询请求 Q，若 u 所在节点的 cnode. Num 大于等于 u 的隐私要求 k，那么就将该节点对应的象限作为查询请求 Q 的隐匿区域；若 u 所在节点的 cnode. Num 小于 u 的隐私要求 k，则检查 cnode 的父节点 cnode - > parent. Num 是否大于等于 k，若是，则将父节点对应的象限作为 Q 的隐匿区域，若不是，再对祖父节点进行类似考查，直到满足隐私需求为止. 为了避免上述的推断攻击，在找到查询 Q 所对应的隐匿区域后，需要将该区域节点的父节点及其它祖先节点的用户数均做减一处理.

如图 3 所示，父节点对应的区域为 <(0, 0), (2, 2)>，四个儿子结点对应的四个象限分别为 <(0, 0), (1, 1)>，<(0, 1), (1, 2)>，<(1, 0), (2, 1)>，<(1, 1), (2, 2)>. 假设用户 C 发出查询请求，且隐私要求 k = 2. C 位于象限 <(0, 0), (1, 1)>，判断该象限中用户数 cnode. Num = 2 ≥ 2，所以根据算法思想，将节点区域 <(0, 0), (1, 1)> 作为用户 C 的隐匿区域，以满足位置的 2 - 匿名，同时也满足了查询的 2 - 匿名. 如果此时用户 A 也发出查询请求 Q，且隐私要求 k = 8，用户 A 所在的象限 <(0, 1), (1, 2)> 中的用户数 cnode. Num = 2 < 8，所以将其父节点区域 <(0, 0), (2, 2)> 作为其隐匿区域，但是根据前面的分析，以 <(0, 0), (2, 2)> 作为其隐匿区域只能保证位置 8 - 匿名，但是不能保证查询 8 - 匿名，查询受到攻击的概率大于 1/8. 按照本文的算法思想，在为用户 C 形成隐匿区域后，还需要将其父节点区域，即 <(0, 0), (2, 2)> 中的用户数减一，得到 cnode - > parent. Num - 1 = 7，因此再为用户 A 寻找隐匿区域时，<(0, 0), (2, 2)> 也不再满足隐私需求 k = 8，

4.2 算法描述

算法具体过程的伪代码如下：

算法 1：防止推断攻击的匿名算法

```
the all query at the time t
for each query    do
      if (Q is not be anonymized)
          while(Q - > cnode. Num < k) //查询用户所在的节点中包含的用户数目小于 k
               Q - > cnode←(Q - > cnode) - > parent
          end while
          if(Q - > cnode. Num ≥ k)
               Q - > anoynode←Q - > cnode
               while((Q - > anoynode) - > parent) / /父节点以及祖先节点的用户数目均减一
                    (Q - > anoynode) - > parent. Num - -
          end while
          end if
    end if
     Qset←Qset - Q
end for
```

4.3 改进算法描述

根据算法 1 的描述，如果查询用户所在的节点 cnode 上用户数 cnode. Num 不能满足用户的隐私需求，那么对该节点的父节点进行验证. 根据四分树的构造，父节点对应的区域面积是其儿子节点对应区域面积的四倍，直接用父节点代替儿子节点作为查询用户的隐匿区域，将使得隐匿区域的面积较大，且空间间隔粒度很大. 如图 3 所示，用户 B 的隐私需求 k = 4 时，用户 B 所在的象限 <(0, 1), (1, 2)> 中的用户数不满足隐私需求，根据前面的算法将对其父节点，即区域 <(0, 0), (2, 2)> 进行验证，父节点满足隐私需求，将区域 <(0, 0), (2, 2)> 作为用户 B 的隐匿区域. 很显然，只要以区域 <(0, 1), (2, 2)> 作

为 B 的隐匿区域就可以满足隐私需求 $k=4$ 了，区域 <(0，1)，(2，2)> 的面积是区域 <(0，0)，(2，2)> 面积的一半. 所以，为了有效的减小隐匿区域的大小，并降低空间的间隔粒度，采用二分树数据结构对算法 1 进行改进，二分树的非叶子节点都有两个儿子，且儿子节点对应的区域为其父节点区域的一半，把这类节点称作半象限节点，而每个半象限节点有两个儿子节点，这种结构等价于在四分树结构的父子节点之间加入了一层半象限节点. 因此，若对同一区域进行划分，采用二分树结构的树高是采用四分树结构下树高的 2 倍.

5　实验结果与分析

实验硬件环境：Intel(R) Core(TM)2 Duo CPU T6600 @2.2GHz，2.7GHz；2GB 内存. 操作系统：Window XP SP2. 编程环境：Visual C + + 6.0.

以城市 Oldenburg 的真实交通路网(面积大约为 25 km × 25 km)作为实验的数据集，并利用 Thomas Brinkhoff[11] 网络移动对象数据生成器生成模拟移动对象数据. 实验中设置移动用户数量的变化范围为 10 ~ 50 K，并从这些移动用户中随机产生 2000 个用户的查询请求，且发出请求的用户，其隐私需求的变化范围为[1 - 20] - [80 - 100]. 表 1 给出了实验中各参数的取值范围和默认值.

表 1　参数的取值范围和默认值

Tab. 1　The scope and default values of different parameters

参数	取值范围	默认值
移动用户的数量	10k - 50k	30k
用户隐私需求 k	[1 - 20] - [80 - 100]	1 - 50
查询请求用户的数量	2k	2k
最小单元格大小/m^2	100 × 100	100 × 100

在实际情况下，同一时刻，不同用户的隐私需求会因各种原因而有所不同，因此，基本算法和改进算法均考虑了个性化的隐私需求. 前述文献中的 Casper 算法和 Interval - Cloak 算法也考虑了个性化的隐私需求，但是这些算法可能存在推断攻击. 算法在考虑个性化隐私需求前提下可以有效的避免推断攻击. 实验将从数据的可扩展性和用户的隐私需求两个方面对算法的执行性能进行评估，并与 Casper 算法和 Interval - Cloak 算法进行比较. 实验中，将采用四分树结构的隐匿算法标记为 Basic，采用二分树结构的改进算法标记为 Adaptive，Interval - Cloak 算法记为 IC 算法.

服务提供商根据匿名位置进行查询处理的代价，以及匿名服务器和服务提供商之间的通信代价都将受到隐匿区域面积大小的影响，因此，采用平均隐匿区域面积大小作为实验的性能指标，用平均隐匿区域面积与整个空间区域面积的百分比来表示. 由于算法在形成隐匿区域的过程中对父节点用户数进行了减一调整，所以理论上 Basic 算法和 Adaptive 算法相对于 Casper 算法和 Interval - Cloak 算法，对同一查询用户会形成更大的空间隐匿区域.

5.1　数据的可扩展性

在整个空间区域面积保持不变的情况下，随着移动用户数的增加，用户密度也随之增加. 在相同隐私需求下，隐匿区域的大小相对减小. 如图 4 所示，四种隐匿算法的平均隐匿区域面积大小都随着移动用户数量的增加而减少. 随着移动用户数量增加，Adaptive 算法和 Casper 算法所产生的平均隐匿区域面积大小逐渐接近；当移动用户数量增加到 30 k 时，Basic 算法和 Interval Cloak 算法形成的平均隐匿区域面积大小基本相等. 这说明本文提出的算法既能防止推断攻击，同时也能获得与之前文献中算法相当的性能.

5.2　隐私需求对算法的影响

众所周知，在 k - 匿名模型下，用户的隐私需求通常用 k 的大小来表示，k 越大，隐私需求越高. 在整个空间区域面积和用户数量保持不变，即网络密度保持不变的情况下，k 值越大，形成的隐匿区域也越大. 如图 5 所示，随着隐私需求 k 的增大，四种不同的隐匿算法产生的平均隐匿区域面积都随之增大. 随着 k 的增大，Basic 算法和 Interval Cloak 算法形成的隐匿区域的大小几乎相等. 由于 Casper 算法和 Adap-

tive 算法均采用相似的半象限处理方法，所以这两种算法形成的平均隐匿区域面积相对于 Basic 算法和 Interval Cloak 算法有所减小，且随着 k 的增大，这种减小的比例愈发明显. 由此可见，Adaptive 算法中引入的半象限节点有效的提升了算法的性能.

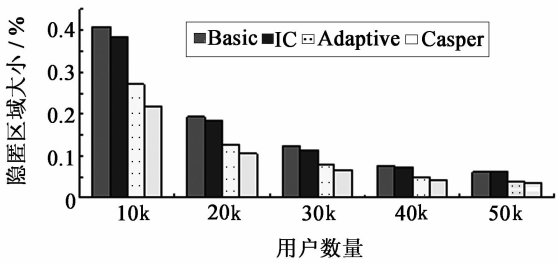


图 4 数据的可扩展性

Fig. 4 The scalability of data

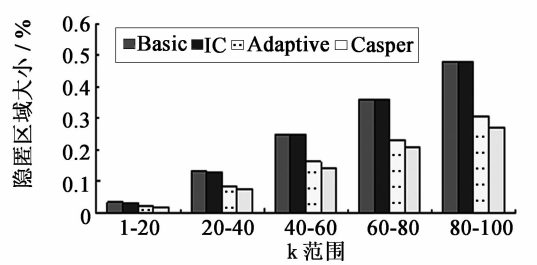


图 5 不同 k 值下隐匿区域面积的变化

Fig. 5 Changes of cloaking area by different k values

6 结语

结合实际情况，分析了现有位置隐私保护算法在用户位置信息公开的前提下，可能造成查询隐私的推断攻击. 提出了两种分别基于四分树结构和二分树结构的隐匿算法，这两种算法在防止攻击者推断攻击的同时还支持个性化的用户隐私需求. 实验结果验证了两种算法的可行性和有效性.

参考文献：

[1] 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] Beresford A R，Stajano F. Location privacy in pervasive computing[J]. IEEE Pervasive Computing，2003，2(1)：46 – 55.

[3] Chow C Y，Mokbel M F. Enabling privacy continuous queries for revealed user locations[C]// Proceedings of International Symposium on Advances in Spatial and Temporal Databases. Boston：Springer，2007：258 – 275.

[4] Gruteser M，Grunwald D. Anonymous usage of location – based services through spatial and temporal cloaking[C]//Proceedings of the International Conference on Mobile Systems，Applications，and Services. NewYork：ACM，2003：31 – 42.

[5] Gedik B，Liu ling. Location privacy in mobile systems：a personalized anonymization model[C]//Proceedings of the International Conference on Distributed Computing Systems. Columbus：IEEE，2005：620 – 629.

[6] Mokbel M F，Chow C Y，Aref W G. The new casper：query processing for location services without compromising privacy[C]// Proceedings of the International Conference on Very Large Data Bases. Seoul：ACM，2006：763 – 774.

[7] Kalnis P，Ghinita G，Mouratidis K，*et al*. Preventing location – based identity inference in anonymous spatial queries[J]. IEEE Transactions on Knowledge and Data Engineering，2007，19(12)：1 719 – 1 733.

[8] Chow C Y，Mokbel M F，Liu X. A peer – to – peer spatial cloaking algorithm for anonymous location – based services[C]// Proceedings of the ACM Symposium on Advances in Geographic Information Systems. New York：ACM，2006：171 – 178.

[9] Ghinita G，Kalnis P，Skiadopoulos S. PRIVE：anonymous location based queries in distributed mobile systems[C]//Proceedings of International Conference on World Wide Web. New York：ACM，2007：1 – 10.

[10] Deutsch A，Hull R，Vyas A，*et al*. Policy – aware sender anonymity in location based services[C]// Proceedings of the 26th IEEE International Conference on Data Engineering. Long Beach：IEEE，2010：133 – 144.

[11] Brinkhoff T. A framework for generating network – based moving objects[J]. GeoInformatica，2002，6(2)：153 – 180.

(责任编辑：林晓)