基于GeoHash的近邻查询位置隐私保护方法

周艺华 李广辉 杨宇光 侍伟敏
(北京工业大学信息学部 北京 100124)

摘 要 随着移动应用和定位技术的不断发展,基于位置的服务(Location-Based Services,LBS)得到了越来越广泛的应用。LBS在为人们提供便利的同时也带来了隐私泄露的风险。近年来,位置服务中的隐私保护问题得到了研究者的持续关注,特别是近邻查询中的位置隐私保护问题得到了广泛的研究。针对第三方匿名服务器缺乏可信性以及容易成为系统瓶颈的问题,提出了一种自适应位置隐私保护强度的不依赖于第三方匿名服务器的基于GeoHash的近邻查询位置隐私保护方法。该方法利用GeoHash算法对用户精确的位置坐标进行字符串编码,将二维经纬坐标转换为一维字符串;LBS服务器通过构建Trie前缀树对GeoHash编码的字符串进行匹配并将查询结果返回给用户。理论分析和实验结果表明,该算法降低了查询通讯开销,同时能够有效保护用户的位置隐私信息。

关键词 位置隐私,基于位置的服务,字符串编码,GeoHash,Trie
中图法分类号 TP309 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2019.08.035

Location Privacy Preserving Nearest Neighbor Querying Based on GeoHash

ZHOU Yi-hua LI Guang-hui YANG Yu-guang SHI Wei-min
(Faculty of Information Technology,Beijing University of Technology,Beijing 100124,China)

Abstract With the continuous development of mobile applications and location technologies,Location-Based Services has become more and more widely used. LBS brings convenience to people while also posing a risk of privacy breaches. In recent years,the issue of privacy protection in location services has received continuous attention from researchers,especially the issue of location privacy protection in neighboring queries has been extensively studied. Aiming at the lack of credibility of third-party anonymous servers and the problem of being a system bottleneck,this paper proposed a GeoHash-based neighbor query location privacy protection method that does not depend on third-party anonymous servers for adaptive location privacy protection. The method uses the GeoHash algorithm to encode the exact position coordinates of the user and convert the two-dimensional latitude and longitude coordinates into a one-dimensional string. The LBS server matches the GeoHash encoded string by constructing the Trie prefix tree and returns the query result to the user. Theoretical analysis and experimental results show that the algorithm reduces the query communication overhead and can effectively protect the user's location privacy information.

Keywords Location privacy,Location-Based services,String encoding,GeoHash,Trie

1 引言

近年来,空间定位以及无线通信技术的发展促进了基于位置的服务的出现。用户通过移动通讯设备向LBS位置服务器发送自身的位置信息来获取相应的基于位置的服务。近邻(Nearest Neighbors,NN)查询是LBS领域中最常见的查询方式,查询过程除了包含用户感兴趣的点(Points of Interest,POI)的相关信息外,还包含用户的精确位置信息。用户发送当前的精确位置给LBS服务器来获取相应的查询服务将不可避免地造成用户位置隐私信息的泄露[1-2]。在近邻查询中,"使用位置服务"和"保护用户位置隐私"是一对必然存在的矛盾。用户只能向LBS位置服务提供方发送模糊化的位置信息,而不能发送自身精确的位置信息,从而实现对用户位置隐私信息的保护。LBS位置服务提供方根据用户提供的模糊位置信息进行近邻查询,然后向用户返回查询结果。随着用户对个体隐私信息安全的持续关注,如何在保证用户位置隐私安全的前提下提供安全可靠的LBS服务成为隐私保护领域研究的热点。

为了有效保护用户的位置隐私信息,文献[3-5]提出了基于可信第三方匿名服务器的位置隐私保护方法,用户将其准确位置信息发送给第三方匿名服器,借由第三方的匿名化处理算法与LBS服务器交互,从而实现对用户位置隐私信息的保护。但是基于可信第三方匿名服务器的方法存在以下不足:1)第三方匿名服务器的安全性难以保证;2)第三方匿名服务器容易成为整个系统的瓶颈。可信第三方匿名服务器位于用户与LBS服务器之间,大量的用户请求被传递给可信第

到稿日期:2018-08-25 返修日期:2018-11-25 本文受北京市自然科学基金项目(4182006),国家自然科学基金项目(61572053)资助。
周艺华(1969-),男,副教授,主要研究方向为网络与信息安全,E-mail:zhouyh@bjut.edu.cn(通信作者);**李广辉**(1992-),男,硕士生,主要研究方向为内容安全,E-mail:n3verl4nd@qq.com;**杨宇光**(1976-),女,教授,主要研究方向为信息安全及信息安全与其他学科的交叉学科;**侍伟敏**(1978-),主要研究方向为网络与信息安全、密码学以及信息安全与其他学科的交叉学科。

三方匿名服务器。与此同时，可信匿名服务器还负责匿名区域的产生，同样需要消耗大量的计算资源。

针对以上问题，本文提出了一种不依赖可信第三方匿名服务器的保护用户位置隐私信息的近邻查询算法 GHNNQ (GeoHash Nearest Neighbor Querying)，即客户端向 LBS 服务器发送经过 GeoHash 编码的用户位置数据，通过在服务器端配置相应的查询处理算法，来实现用户与 LBS 位置服务器的直接交互。此处，GeoHash 编码起到了对用户精确位置模糊化的作用，从而实现了对用户位置隐私信息的保护。

2　相关工作

基于位置服务的隐私保护研究根据保护对象的不同分为针对查询内容的隐私保护和针对查询位置的隐私保护。根据是否需要可信第三方匿名服务器，其分为基于可信第三方模式(Trusted Third Party，TTP)和不依赖可信第三方模式。1)基于可信第三方模式，客户端将用户查询内容和精确位置信息发送给可信第三方匿名服务器，可信第三方匿名服务器负责隐藏用户的精确位置信息，并通过与 LBS 位置服务器的交互来获得查询结果并将其返回给用户；2)不依赖可信第三方模式，用户直接与 LBS 位置服务器通信，并完成相关的查询请求。

保护位置隐私近邻查询的策略是：对查询者的精确位置信息进行模糊化，将模糊化后的位置信息发送给 LBS 服务器进行处理并将查询结果返回给用户。针对 LBS 领域中的关于用户的近邻查询位置隐私安全问题，研究者已经提出了多种位置隐私保护方法，主要有假位置法、空间转换法和空间匿名法 3 类。

Gkovlalasdivanis 等[5-8] 提出了一种基于 K 匿名模型的 LBS 位置隐私保护方法，其核心思想是将包含 K 个用户精确位置的查询请求发送给服务器，使攻击者在 K 个无差别的用户位置中无法区分出用户的真实位置。该方法的隐私保护强度依赖于 K 的大小，隐私保护强度不可控并且查询开销大。Yiu 等[9-11] 提出了一种基于假位置的位置隐私保护方法，即客户端向 LBS 服务器发送假位置，LBS 服务器返回关于假位置的查询结果。重复执行这一过程，直到服务器返回的查询结果满足用户的准确性要求。该方法虽然实现过程简单，但是需要连续多次查询，通信开销较大。Khoshgozaran 等[12-13] 提出了一种基于 Hilbert 空间填充曲线的位置隐私保护方法，即使用 Hilbert 空间填充曲线将用户位置从二维空间转换为一维空间，用户的二维精确位置信息被转换为了一维的索引从而实现了对用户位置隐私信息的保护。该方法的计算开销大且准确率低。Ghinita 等[14] 提出了一种基于私有信息检索(Private Information Retrieval，PIR)的位置隐私保护方法，即采用二次剩余原理构建寻找最近邻的方法。该方法虽然隐私保护强度高，但是计算及通信开销大。

3　系统架构

本文提出的算法的系统架构如图 1 所示。整个系统由 3 个部分组成：移动用户、负载均衡服务器和位置服务提供商的服务器。移动用户为 LBS 服务的请求者，负载均衡服务器根据移动用户的请求选择合适的 LBS 服务器，位置服务提供商为 LBS 服务的提供者。

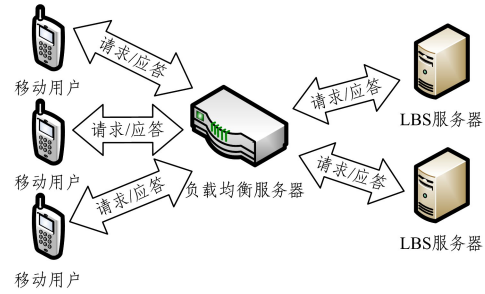

图 1　系统架构

Fig. 1　System structure

用户使用移动终端向 LBS 服务器发送请求，并获得最终的查询结果。为了提高查询效率，本文使用负载均衡服务器，避免用户直接与 LBS 服务器交互。移动用户将编码后的位置信息发送给负载均衡服务器，负载均衡服务器根据编码后的字符串请求相应的 LBS 服务器并获得查询结果。此处，LBS 服务器作为一个集群，包含多台服务器，每台服务器含有各自不同的 POI 数据集。负载均衡服务器将不同的查询请求分发给不同的 LBS 服务器，从而提高整个系统的查询处理效率。

为了避免对数据库进行频繁访问，提高后端服务的响应速度，在负载均衡服务器中部署缓存模块。当用户需要读取数据时，会先从缓存中查找所需的数据，如果找到则直接返回查询结果，否则从数据库中读取。

4　算法实现

GHNNQ 算法的实现包括以下两个部分：客户端算法对用户坐标进行 GeoHash 编码，用来实现对用户真实位置的隐藏；服务端算法根据客户端请求进行 Trie 树查找，并将查询结果返回给客户端。

4.1　客户端算法

常见的保护位置隐私的方法有：假位置法、空间匿名法和空间转换法，本文采用的是空间匿名法。空间匿名的基本思想是：用一个包含查询者当前精确位置的区域代替查询者的位置坐标，向 LBS 位置服务器发起关于该区域的近邻查询请求。如图 2 所示，用户 e 的真实位置坐标为 (lat, lng)，空间匿名方法就是将此点扩充为图 2 中的矩形匿名区域，即用这个矩形区域代表用户 e 的真实位置。

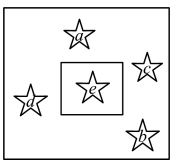

图 2　空间匿名

Fig. 2　Space anonym

由于用户在此矩形区域内的每一个位置出现的概率相同，因此攻击者仅能够知道用户在这个矩形区域内，但无法确定用户是在整个区域内的哪个具体位置。

匿名区域的大小反映了匿名强度。匿名区域越大，可能覆盖的用户数越多，匿名的效果就越好，但是查询处理效率就越低；匿名区域越小，匿名效果就越差，但是查询处理效率就越高。

为了确保查询用户的位置隐私安全，客户端不能将用户的具体位置信息直接发送给 LBS 服务器，而需要对查询用户的位置信息进行匿名化处理，同时要求 LBS 服务器能够根据匿名后的位置信息将准确的查询结果反馈给用户。通过 GeoHash 算法将用户的二维经纬坐标转换成一个可排序、可比较的字符串编码序列，然后将该序列发送给 LBS 服务器，避免直接发送用户的精确位置信息。编码后的每个字符串代表一个矩形区域，并且前缀字符串是后面字符的父区域。

已知地球纬度区间是[－90，90]，经度区间是[－180，180]。下面以纽约自由女神像的经纬度坐标(40.689 249，－74.046 689)为例，介绍 GeoHash 编码算法的实现过程。

对于纬度坐标 40.689 249，GeoHash 编码的过程如下：

对纬度区间[－90，90]进行二次划分，从而形成两个子区间[－90，0]和[0，90]，称其为左右区间，可以确定 40.689 249 属于右区间[0，90]，将其标记为 1；

对纬度区间[0，90]进行二次划分，从而形成两个子区间[0，45)和[45，90]，可以确定 40.689 249 属于左区间 [0，45]，将其标记为 0；

重复执行上述查找过程，纬度 40.689 249 总是属于某个区间[x，y]。随着迭代区间[x，y]的缩小，该区间会越来越逼近 40.689 249。

如表 1 所列，如果经纬坐标属于右区间则标记为 1，这样随着算法的进行会产生一个二进制编码序列：10111 00111。序列的长度与给定的区间划分次数有关，二次划分次数越多，二进制编码序列就越长。

同理，对经度－74.046 689 进行 GeoHash 编码的过程也类似。

如表 2 所列，经度坐标－74.046 689 产生的二进制编码序列为：01001 01101。

通过上述计算，纬度产生的二进制编码序列为 10111 00111，经度产生的二进制编码序列为 01001 01101。在偶数位置(索引从 0 开始)存放经度，在奇数位置存放纬度，把两个二进制编码序列合并成一个新的二进制编码序列 01100 10111 00101 10111。从右向左，每 5 位为 1 组，不够 5 位的左边补 0。

表 1 纬度编码

Table 1 Latitude encoding

纬度范围	划分区间 0	划分区间 1	39.87454 所属区间
[－90，90]	[－90，00]	[0.0，90]	1
[0.0，90]	[0.0，45.0]	[45.0，90]	0
[0.0，45.0]	[0.0，22.5]	[22.5，45.0]	1
[22.5，45.0]	[22.5，33.75]	[33.75，45.0]	1
[33.75，45.0]	[33.75，39.375]	[39.375，45.0]	1
[39.375，45.0]	[39.375，42.1875]	[42.1875，45.0]	0
[39.375，42.1875]	[39.375，40.7812]	[40.7812，42.1875]	0
[39.375，40.7812]	[39.375，40.0781]	[40.0781，40.7812]	1
[40.0781，40.7812]	[40.0781，40.4296]	[40.4296，40.7812]	1
[40.4296，40.7812]	[40.4296，40.6054]	[40.6054，40.7812]	1

表 2 经度编码

Table 2 Longitude encoding

经度范围	划分区间 0	划分区间 1	－74.046 689 所属区间
[－180，180]	[－180，0.0]	[0.0，180]	0
[－180，0.0]	[－180，－90.0]	[－90.0，0.0]	1
[－90.0，0.0]	[－90.0，－45.0]	[－45.0，0.0]	0
[－90.0，－45.0]	[－90.0，－67.5]	[－67.5，－45.0]	0
[－90.0，－67.5]	[－90.0，－78.75]	[－78.75，－67.5]	1
[－78.75，－67.5]	[－78.75，－73.125]	[－73.125，－67.5]	0
[－78.75，－73.125]	[－78.75，－75.9375]	[－75.9375，－73.125]	1
[－75.9375，－73.125]	[－75.9375，－74.5312]	[－74.5312，－73.125]	1
[－74.5312，－73.125]	[－74.5312，－73.8281]	[－73.8181，－73.125]	0
[－74.5312，－73.8281]	[－74.5312，－74.1796]	[－74.1796，－73.8281]	1

最后使用 0－9、a－z(去掉容易混淆的字符 a，i，l，o)32 个字母进行 Base32 编码(见表 3)，首先将 01100 10111 00101 10111 转成十进制 数 12，23，5，23。以上 8 个十进制数字对应的 Base32 编码后的字符串就是 dr5r。该字符串为(40.689 249，－74.046 689)对应 GeoHash 编码后的字符串。

表 3 Base32 编码表

Table 3 Base32 alphabet

十进制	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Base32	0	1	2	3	4	5	6	7	8	9	b	c	d	e	f	g	h	j	k	m	n	p	q	r	s	t	u	v	w	x	y	z

在纬度相等的情况下，经度每隔 0.000 01 度，距离相差约 1 m；每隔 0.000 1 度，距离相差约 10 m；每隔 0.001 度，距离相差约 100 m；每隔 0.01 度，距离相差约 1 000 m；每隔 0.1 度，距离相差约 10 000 m。在经度相等的情况下：纬度每隔 0.000 01 度，距离相差约 1.1 m；每隔 0.000 1 度，距离相差约 11 m；每隔 0.001 度，距离相差约 111 m；每隔 0.01 度，距离相差约 1 113 m；每隔 0.1 度，距离相差约 11 132 m。由此可以得到 GeoHash 编码的精度，如表 4 所列。

由表 4 可知，字符串编码越长，其表示的范围就越精确。当 GeoHash 编码字符串的长度为 8 时，精度约为 19 m，而当编码长度为 9 时，精度约为 2 m。编码长度需要根据位置隐私保护强度进行选择，同时还需要考虑查询效率。GeoHash 用一个字符串表示经度和纬度两个坐标，它表示的是一个矩形区域，矩形区域的每一个点都有可能是用户的精确位置，从而使攻击者无法判断出用户的确切位置。经过 GeoHash 编码的字符串越长，矩形区域就越小，结果就越准确，但是隐私保护强度就越弱。由于在客户端实现 GeoHash 字符串编码算法，因此隐私保护强度可由用户来进行控制。

表 4 Base32 精度

Table 4 Base32 precision

Geohash length	Lat bits	Lng bits	Lat error	Lng error	Km error
1	2	3	±23	±23	±2 500
2	5	5	±2.8	±5.6	±630
3	7	8	±0.70	±0.70	±78
4	10	10	±0.087	±0.18	±20
5	12	13	±0.022	±0.022	±2.4
6	15	15	±0.002 7	±0.005 5	±0.61
7	17	18	±0.000 68	±0.000 68	±0.076
8	20	20	±0.000 085	±0.000 17	±0.019

但是 GeoHash 编码算法存在一个缺点，虽然位于边界两侧的两个 POI 的距离十分接近，但编码的字符串会完全不同。在实际应用中，同时搜索该点所在区域的其他 8 个相邻区域内的 POI，即可解决这个问题。

4.2　服务端算法

服务器端存储了查询对象的 POI 信息，例如某区域内所有酒店的二维经纬坐标地址以及使用 GeoHash 编码后的字符串。客户端仅提交对其坐标进行 GeoHash 编码后的字符串，在服务器端执行查询操作并将结果返回给客户端。服务端算法采用 Trie 树实现。

如图 3 所示，该多叉树就是一棵 Trie 树，用 10 个结点保存了 7 个字符串{tea，ten，to，in，inn，int}以及两个字符{i，t}。在该 trie 树中，字符串 in，inn 和 int 的公共前缀是 in，因此可以只存储一份 in 以节省空间。

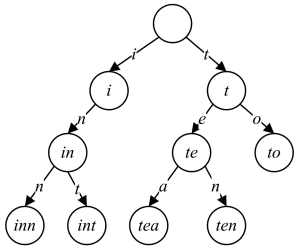

图 3　Trie 树

Fig. 3　Trie tree

根据图 3 可以归纳出 Trie 树的一些特点：

1）根结点不包含任何字符，除根结点外的每一个子孙结点都包含一个字符并且有唯一的父结点。

2）从根结点到某一个子孙结点，将路径上经过的字符连接起来，即为该结点对应的字符串。

3）每个结点的所有子结点包含的字符互不相同。

与哈希表不同的是，Trie 树中不同的关键字不会产生冲突，并且查询的效率很高，时间复杂度为 $O(n)$，其中 n 是待查询的字符串的长度。

假设 GeoHash 算法编码后的字符串长度为 9，那么最多只需要 9 次查询就可以判断客户端发送的字符串与数据库中的字符串是否具有相同的前缀，其精度可以保持在 2 m 左右，能够满足我们的实际需求，并且字符串只包含 32 个字符，可以很容易地构建出 Trie 树。

Trie 树结点的子结点的数量固定为 32，可以使用固定长度的数组来保存结点的子结点，其优点是对子结点进行查询时速度较快，缺点是浪费空间，不管有多少子结点都得分配 32 个存储空间。使用链表保存结点避免了空间浪费，但增加了查询时间复杂度。Double-Array 结合数组查询效率高、链表节省空间的优点，使用两个数组 base 和 check 来实现。base 数组中的元素与 Trie 树中的结点一一对应。base 数组中的元素作为状态转移的基值，check 数组中的元素作为校验值，用于检查该状态转移是否存在。从状态 R 到状态 S 的一个转移必须满足如下转移方程：$base[R]+c=S$；$check[S]=R$，其中 c 是待输入变量。

假设数组下标索引为 i，$base[i]$的值为负数时表示该状态是一个结束状态；$base[i]$和 $check[i]$的值均为 0 时表示该状态不存在。

$base[]$和 $check[]$两个数组的构造过程如下：对于状态 $ST_1, ST_2, ST_3, \cdots, ST_n$，状态 ST 在数组中的下标为 i，令 $base[i]=k$；$check[i]=0$，k 满足条件：

$$base[k+T_1]=0, check[k+T_1]=0;$$

$$base[k+T_2]=0, check[k+T_2]=0;$$

$$\cdots$$

$$base[k+T_n]=0, check[k+T_n]=0$$

k 值确定以后，状态 $ST_1, ST_2, ST_3, \cdots, ST_n$ 在 check 数组中的下标随即确定，分别为 $k+T_1, k+T_2, k+T_3, \cdots, k+T_n$，同时需要满足 $check[k+T_1]=check[k+T_2]=\cdots=check[k+T_n]=i$。

构造完数组后需要查询一个结点 ST 是否存在，只需要判断 $check[base[S]+T]$的值是否等于 S。如果相等，则表示 ST 在 Trie 树中能够检索到，随即可以查找所有以 ST 为前缀的子结点；否则，返回没有该关键字。

5　实验和结果分析

本节对 GHNNQ 算法的有效性进行了验证。首先通过更改处理 GHNNQ 算法的参数来验证该算法的实际可行性，然后在不同规模的数据集上与 SpaceTwist 方法进行比较。

本算法采用 Java 以及 MATLAB 实现，计算机的配置如下：Intel(R) Core(TM) i5-6300HQ CPU @ 2.30 GHz，8 GB 内存，操作系统为 Win10。实验数据来自于中国某几个省份的 POI 数据集，一共包含约 400 万条，并存储于 MySQL 数据库中，实验数据如表 5 所列。

表 5　实验数据

Table 5　Experimental data

id	name	address	lat	lng	geohash
1	天缘餐厅	房山区涞宝路	39.6326899	115.59439531	wx41qqvm0
2	十渡	近郊房山区十渡镇拒马河畔	39.63478	115.59796	wx41qrrcg
3	八达岭山洞福利展览馆	北京市昌平区石佛寺	39.45301	116.41168	wx4b3n0qk
4	恋日绿岛	朝阳区东南四环外博大路西侧	39.828749	116.488495	wx4fdf876
…	…	…	…	…	…

如图 4 所示，匿名空间面积随着 GeoHash 编码长度 L 的增大而减小，L 值越大，相应的匿名空间越小，隐私保护强度就越弱。

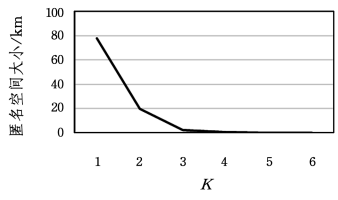

图 4　匿名空间大小

Fig. 4　Anonymous space size

图 5 给出了使用 GeoHash 编码以及 Trie 树进行查询时响应时间的差异。可以看到，使用 GeoHash 编码能够明显提

高查找的效率。在服务器端使用 Trie 前缀树同样能够显著提高查询效率，并且在数据集增大的情况下基本保持固定的查询处理时间。

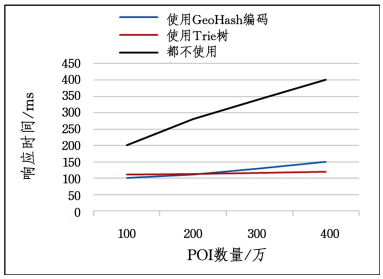

图 5 算法的处理时间(电子版为彩色)

Fig. 5 Processing time of algorithm

实验也将 GHNNQ 方法与同样无第三方匿名服务器的 SpaceTwist 算法进行了比较。如图 6 所示，在相同的隐私保护强度下，随着 POI 数量的增加，两种方法的平均处理时间都有所增长。相比 SpaceTwist 增量查询方法，GHNNQ 方法在处理时间上有明显的优势。

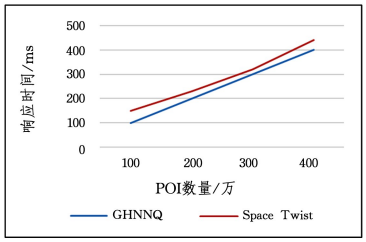

图 6 算法处理时间的性能比较(电子版为彩色)

Fig. 6 Algorithm processing time performance comparison

结束语 针对现有的基于可信第三方匿名服务器的保护位置隐私的近邻查询算法的不足，本文提出了一种不依赖于可信第三方匿名服务器的基于 GeoHash 编码的保护用户位置隐私信息的近邻查询方法(GHNNQ)。由于采用了 GeoHash 编码，把用户精确的二维经纬坐标转换为一维的字符串，每个编码过后的字符串代表一个包含用户当前位置的区域，达到了模糊用户精确位置的效果，从而实现了对用户位置隐私的保护。LBS 服务器通过设计针对 GeoHash 编码字符串的查询处理算法，将包含查询结果的候选集反馈给客户端。另外，即使用户以及 POI 经纬坐标发生微小的变化，经纬坐标也能编码成相同的 GeoHash 字符串，这就保证了在每次执行相同的 Tire 树查询的同时也提高了缓存命中率。本架构在保护用户位置隐私信息的同时，并没有给系统增加额外的负担，相反由于使用了负载均衡和 Trie 查找树，提高了系统的整体性能。本文在真实的全国 POI 数据集上进行了充分的实验，证明了该方法的优越性。

参 考 文 献

[1] WANG L, MENG X F. Location privacy preservation in big data era: A survey[J]. Journal of Software, 2014, 25(4): 693-712. (in Chinese)
王璐，孟小峰. 位置大数据隐私保护研究综述[J]. 软件学报，2014，25(4)：693-712.

[2] ZHANG X J, GUI X L, WU Z D. Privacy preservation for location-based services: A survey[J]. Journal of Software, 2015, 26(9): 2373-2395. (in Chinese)
张学军，桂小林，伍忠东. 位置服务隐私保护研究综述[J]. 软件学报，2015，26(9)：2373-2395.

[3] GEDIK B, LIU L. Protecting location privacy with personalized k-anonymity: Architecture and algorithms[J]. IEEE Transaction on Mobile Computing, 2008, 7(1): 1-18.

[4] GEDIK B, LIU L. A customizable k-anonymity model for protecting location privacy[C]// Proceedings of the IEEE International conference on Distributed Computing System(ICDS'05). 2005: 620-629.

[5] GKOULALASDIVANIS A, KALNIS P, VERYKIOS V S. Providing K-Anonymity in location based services[M]. ACM, 2010.

[6] BAMBA B, LIU L, PESTI P, et al. Supporting anonymous location queries in mobile environments with privacy grid[C]// International World Wide Web Conference WWW. 2008: 237-246.

[7] CHOW C Y, MOKBEL M F, LIU X. A peer-to-peer spatial cloaking algorithm for anonymous location-based service[C]// Acm International Symposium on Advances in Geographic Information Systems. ACM, 2006.

[8] PARK G G. MobiHide: A Mobilea Peer-to-Peer System for Anonymous Location-Based Queries[M]// Advances in Spatial and Temporal Databases. DBLP, 2007.

[9] KIDO H, YANAGISAWA Y, SATOH T. An anonymous communication technique using dummies for location-based services[C]// ICPS'05. Proceedings. International Conference on Pervasive Services, 2005. IEEE, 2005.

[10] 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]// IEEE 24th International Conference on Data Engineering, 2008. ICDE 2008. IEEE, 2008.

[11] MOKBEL M F, CHOW C Y, AREF W G. The New Casper: Query Processing for Location Services without Compromising Privacy[C]// Proceedings of the 32nd International Conference on Very Large Data Bases. Seoul, Korea, 2006.

[12] KHOSHGOZARAN A, SHAHABI C, SHIRANI-MEHR H. Location privacy: going beyond K-anonymity, cloaking and anonymizers[M]. Springer-Verlag New York, Inc. 2011.

[13] KHOSHGOZARAN A, SHAHABI C. Blind evaluation of nearest neighbor queries using space transformation to preserve location privacy[C]// SSTD'07 Proceedings of the 10th international conference on Advances in spatial and temporal databases. 2007: 239-257.

[14] GHINITA G, KALNIS P, KHOSHGOZARAN A, et al. Private queries in location based services: Anonymizers are not necessary[C]// Proceedings of the ACM SIGMOD International Conference on Management of Data. Canada: ACM, 2008.