(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 104080081 A
(43)申请公布日 2014.10.01

(21)申请号 201410267766.6

(22)申请日 2014.06.16

(71)申请人 北京大学
地址 100871 北京市海淀区颐和园路 5 号北京大学

(72)发明人 沈晴霓 韩笑 方跃坚 吴中海

(74)专利代理机构 北京君尚知识产权代理事务所（普通合伙） 11200
代理人 邵可声

(51)Int.Cl.
H04W 12/02(2009.01)

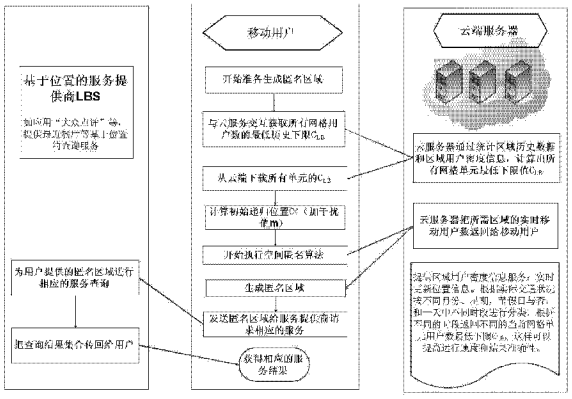
权利要求书1页 说明书13页 附图7页

(54)发明名称

一种适用于移动端位置隐私保护的空间匿名化方法

(57)摘要

本发明公开了一种适用于移动端位置隐私保护的空间匿名化方法。本方法：1）移动端基于位置的查询时，首先向云服务器查询每个网格单元的移动用户数的历史下限值 c_{LB}；2）该移动端根据网格单元的下限值 c_{LB}，对该空间区域进行自下而上的递归，计算出用于实际空间匿名区域计算的初始递归区域；3）该移动端与云服务器交互，从该初始递归区域进行自上而下递归询问云服务器，根据返回的网格单元内的当前用户数量确定是否满足生成匿名区域的条件，如果满足则生成匿名区域，将其发送给基于位置的服务提供方，请求获得相应的服务；4）该基于位置的服务提供方将查询到的消息队列传返回给该移动端。本发明既保证了隐私保护的有效性又降低了通讯成本。



CN 104080081 A

1. 一种适用于移动端位置隐私保护的空间匿名化方法，其步骤为：

1) 移动端采用金字塔型网格数据结构维护空间区域的移动用户信息；当进行基于位置的服务查询时，该移动端首先向云服务器查询该空间区域中每个网格单元的移动用户数的历史下限值 c_{LB}；

2) 该移动端根据返回的所有网格单元的移动用户数的历史下限值 c_{LB}，对该空间区域进行自下而上的递归，计算出用于实际空间匿名区域计算的初始递归区域；

3) 该移动端与云服务器进行交互，从该初始递归区域开始对该空间区域进行自上而下递归，每次递归都要询问云服务器，查询网格单元内的当前移动用户数量，并根据返回的网格单元内的当前移动用户数量，判断是否满足生成匿名区域的条件，如果满足则生成匿名区域，将其发送给基于位置的服务提供方，请求获得相应的服务；

4) 该基于位置的服务提供方将查询到的消息队列传返回给该移动端。

2. 如权利要求 1 所述的方法，其特征在于云服务器通过统计该空间区域的移动用户数的历史数据和用户密度信息，计算出当前该空间区域中所有网格单元的移动用户数的历史下限值 c_{LB}。

3. 如权利要求 2 所述的方法，其特征在于云服务器按不同月份、不同星期、节假日与非节假日和一天中不同时段对统计数据进行分类，根据查询请求所属的时段返回当前网格单元对应的移动用户数的历史下限值 c_{LB}。

4. 如权利要求 1 所述的方法，其特征在于计算出所述初始递归区域的方法为：利用 Casper 匿名算法和每个网格单元的移动用户数的历史下限值 c_{LB}，对该空间区域从最小的网格单元开始进行自下而上的递归，直到计算出该空间区域的第 N 层的一网格单元 c^{LB} 包含该移动端的用户，且其中的移动用户数的历史下限值 c_{LB} 大于或等于该 Casper 匿名算法中的 k- 匿名的设定值 k，将该网格单元 c^{LB} 作为所述初始递归区域。

5. 如权利要求 4 所述的方法，其特征在于设置一干扰值 m，所述设定值 k ＝ k+m。

6. 如权利要求 5 所述的方法，其特征在于所述 m 值的取值范围在 3 ～ 5。

7. 如权利要求 1 所述的方法，其特征在于生成所述匿名区域的方法为：用 c^C 表示当前的网格单元，在递归开始时令 $c^C = c^{LB}$；查询 c^C 的四个子网格单元各自的实时用户数，移动端的用户所在的子网格单元记作 c，如果 c 拥有的用户数不小于设定值 k，则将该子网格单元 c 作为当前单元 c^C，继续进行递归；如果子网格单元 c 的实时用户数小于 k，则检查子网格单元 c 及与其垂直相邻网格单元 c^H 的用户总和，或子网格单元 c 及与其水平相邻网格单元 c^V 的用户总和，如果总和不小于 k，并且该子网格单元 c 的邻居网格单元的当前用户数小于 k，则返回该子网格单元 c 和它的邻居网格单元的区域并集作为所述匿名区域，停止递归；否则将返回当前网格单元作为匿名区域。

8. 如权利要求 7 所述的方法，其特征在于设置一干扰值 m；把计算出的初始递归区域所在的第 N 层，向上扩大 m 层，把该空间区域的第 N-m 层作为初始递归区域所在的层，初始递归区域为第 N-m 层的用户所在的子网格单元。

9. 如权利要求 8 所述的方法，其特征在于所述 m 值的取值范围在 0 ～ 3。

一种适用于移动端位置隐私保护的空间匿名化方法

技术领域

[0001] 本发明涉及一种适用于移动端位置隐私保护的空间匿名化方法，属于网络安全技术领域。

背景技术

[0002] 近年来，随着移动智能手机的高速发展，基于位置的服务 (LBS) 变得越来越流行，这主要是由于现代移动设备集成了位置传感器，从而提高了这种服务的可用性。典型的例子有兴趣查找或者基于物理位置的社交网络，如人人网和 Facebook 的地方，全球卫星定位系统，大众点评应用中根据我的位置搜索附近的餐厅，微信应用里面的显示当前位置信息等。结合用户当前的位置信息，以帮助用户获得更好地服务与便利。但是，提供这种基于用户私人位置的服务可能会引起严重的隐私问题，如果这些位置没有得到充分保护的话，后果将会非常严重，可能导致用户的位置隐私被非法分析或者被连续跟踪利用。因此位置隐私的概念成为一种强制性的概念，以保证移动用户可以接纳基于位置的服务。

[0003] 对于移动用户的位置隐私保护方法在各种文献中已有详细描述。其中，k- 匿名 (k-anonymity) 是一个通用的隐私方法，其思想是用一个至少包含 k 个对象（移动用户）的模糊区域代替请求发起者的准确位置，保证目标对象和其他 k-1 个对象是不可区分的。因此，确定目标用户的概率是 1/k。在此基础上空间匿名化方法被提出和研究，用于保护移动用户在使用基于位置的服务时的位置隐私。传统的空间匿名化方法是要依赖于受信任的代理称为可信的位置服务器 LTS(Location Trusted Server)，用来为每个请求产生一个至少包含 k 个用户的匿名空间。一个典型的空间匿名化方法是 New Casper，Casper 是一个新的框架，在此框架中移动的和静止的用户都可以获取基于位置的服务，同时并不会透露他们的位置信息。由于 Casper 方法的目标是既快且安全地寻找尽可能小的匿名区域，所以 Casper 方法达到了很好的服务品质。但 Casper 也是依赖于单一的 LTS，而假设 LTS 知道所有用户在任何时间的位置信息，为所有发出请求的用户产生匿名空间，所以依赖单一 LTS 进行隐私保护，也面临着一系列的问题，LTS “知道的太多了”，并且单一 LTS 的可扩展性不好，容易成为攻击的焦点。为了改善这种依赖单一服务器的问题，有人提出了借助于云服务的设备端的移动用户隐私保护的空间匿名化方法（下文将简称为设备端的空间匿名化方法）可能是更可取的。但棘手的问题是，手机设备不知道在请求时其他用户的位置信息，而这在产生相应的匿名区域的过程中是必不可少的。随着云服务的发展，可以假设用户密度信息可以从云端服务器获得。这些服务器可以收集不同地区的用户位置信息，也可通过使用复杂的方法来估算不同地方的用户密度，当用户发出一个匿名请求时，移动终端到云端获取对应的用户密度信息来计算空间匿名区域。

[0004] 设备端的空间匿名化方法借助了云服务的支持，可以通过云服务器提供准确的用户密度信息。此方法修改了传统的 Casper 空间匿名算法，把从最小的网格单元开始向上扩大范围的自下而上的递归算法改为从最大的区域开始逐渐向下缩小范围的自上而下的递归结构。但是，从根节点开始递归将增加大量的通讯资费，也会对计算速度造成影响，所以

此方法期望再次改进算法，使用基于用户密度的历史数据的下限值作为初始值，意图改进从根节点递归带来的通信消耗。然而，其改进的过程并没有改善相应的速度问题，并且还会因此带来显著的隐私泄露风险，因为一旦当从云端服务器获取的移动用户数的历史下限值比当前实际用户数大时，将会出现实际区域用户数小于k-匿名的阀值k的情况，那么，k-匿名将失效，发起请求的移动用户位置将暴露。所以，对算法的改进回到了用安全来换取速度和资费的问题。

[0005] 现有技术一的技术方案

[0006] 对于移动用户的位置隐私保护方法在各种文献中已有详细描述。New Casper 空间匿名化方法是一种新的解决隐私泄露模型的框架，在此框架中移动的和静止的用户都可以获取基于位置的服务，同时还可以保护他们的位置信息。Casper 主要由两部分组成 :位置匿名器和隐私感知查询处理器。位置匿名器根据用户指定的隐私需求，把用户的准确位置信息模糊成空间匿名区域。隐私感知查询处理器嵌入于基于位置的数据库服务器中，是为了处理空间匿名区域，而不是准确的位置信息。Casper 使用传统的基于位置的服务器和查询处理器给他们的客户提供匿名服务。移动用户可以使用基于位置的服务，而无需透露自己的私人位置信息。通过 Casper 框架计算出一个拥有 k 个用户的匿名区域，在这个框架中，由用户自己设定 k 的值和用来隐藏用户位置的匿名区域的最小面积 Amin。通过注册 Casper，移动用户可以设置隐私设定档来指定隐私的方便程度。用户隐私的配置文件包括两个参数 k 和 Amin。参数 k 表示请求 k- 匿名的移动用户与其他 k-1 个用户是不可区分的，Amin 表示用户想要隐藏自己的位置信息在一个大小至少为 Amin 的区域内。k 和 Amin 较大的取值表示具有更严格的保密要求。

[0007] 场景描述 :Casper 结构主要包含两个组件 :位置匿名器和隐私感知查询处理器。移动用户向位置匿名器连续发送位置更新信息，位置匿名器产生空间匿名区域来模糊位置更新，以保证与每个移动用户的隐私配置 (k，Amin) 相匹配，并将该匿名空间区域发送到基于位置的数据库服务器。在伪装用户位置信息的同时，匿名器也会删除任何的用户身份信息，以确保位置信息的假名成立。与用户的位置点类似，匿名器也会在发送匿名查询区域给基于位置的数据库服务器之前，先模糊查询的位置信息。

[0008] 隐私感知查询处理器内嵌在基于位置的数据库服务器中，以匿名方式处理空间匿名区域而不是位置点。相对于直接返回一个确切的答案，隐私感知查询处理器返回一个查询候选列表，以答复通过位置匿名器产生的基于位置的查询。移动用户将在本地评估查询后返回的候选列表，找到自己所期望的结果。隐私感知查询处理器可以保证查询候选列表是最小的，并且包含用户所需的答案。候选列表的大小很大程度上取决于用户的隐私配置。严格的隐私配置可能会返回一个较大的候选列表。通过设置隐私配置，移动用户可以自主地在透露自己位置信息的敏感程度和从 Casper 获得的服务品质之间做出一个合理的权衡。基于位置的数据库服务器中处理的位置隐私感知查询，可以是来自移动用户，也可以是公共管理者。来自移动用户的查询被作为私人的查询，需要先通过位置匿名器来隐藏查询的身份信息，并且模糊查询发出者的位置。来自于公共管理者的基于位置的查询被作为公共查询，不需要通过位置匿名器，而是直接把它们提交给基于位置的数据库服务器。数据库服务器将根据所有移动用户存储的匿名位置信息来回答这样的公众查询。

[0009] 现有技术一的缺点 :

[0010] Casper 方法可以既快且安全地寻找尽可能小的匿名区域，可以达到很好的服务品质。但是这样做仍然存在一些问题：

[0011] Casper 是依赖于单一的 LTS，而假设 LTS 知道所有用户在任何时间的位置信息，为所有发出请求的用户产生匿名空间，所以依赖单一 LTS 进行隐私保护，也面临着一系列的问题，LTS “知道的太多了”，并且单一 LTS 的可扩展性不好，容易成为攻击的焦点。

[0012] 现有技术二的技术方案

[0013] 另外一种位置隐私保护方法是现有的借助云服务的设备端的空间匿名化方法。

[0014] 这套方案取消了 LTS，把空间匿名算法放到移动设备端实现，而相关的移动用户位置信息，则借助云服务的帮助，从云服务提供商处获得计算所需相关区域的移动用户密度信息和用户的实时位置信息。这个过程虽然改进了传统的 Casper 空间匿名算法从下到上

[0015] (Down-Top) 的递归算法，变为自上而下 (Top-Down) 的递归，却带了通讯成本和速度的问题，甚至还存在位置隐私泄露的威胁。

[0016] 设备端的空间匿名化方法的具体工作流程如图 1 所示。

[0017] 我们以移动用户的整个使用过程来说明设备端的空间匿名化方法的工作流程，下面详细介绍用户使用过程。

[0018] 第 1 步，移动用户在手机设备上负责产生匿名区域，移动设备客户端通过递归分割空间，区域分区通过索引的方式储存在网格结构上。对于每个网格单元，它可以与云服务器交互，询问在某一时刻这个网格单元内的移动用户数量。

[0019] 第 2 步，云服务器会更新用户的位置信息，随着空间中移动用户不断移动，网格单元内的用户数量随时间而变化。所以对于云服务器来说，要随时获得用户位置信息的更新非常重要。云服务器会提供实时区域用户密度信息。所以云服务器会回传所需区域的移动用户位置的统计信息给移动用户手机设备。

[0020] 第 3 步，移动用户通过从云端收到的实时用户位置和密度信息，独立产生匿名空间。

[0021] 第 4 步，一旦移动用户设备生成一个匿名的区域，它就会把包括生成的匿名区域的请求发送给基于位置的服务提供商 (LBS)，如应用“大众点评”，询问最近的餐厅。

[0022] 第 5 步，LBS 会把查询到的结果集合传回给用户。

[0023] 由此可见，给用户的手机终端造成了计算的压力，并且频繁的与云端交互可能会带来通讯资费的提升。

[0024] 现有技术二的缺点

[0025] 设备端的空间匿名化方法通过与云服务器的结合，避免了 LTS 带来的弊端，可以产生有效的尽可能小的匿名区域。但是也存在一些问题：

[0026] 1. 因为根节点的移动用户数量远远大于 k- 匿名目标值 k，所以把根节点作为起始点始递归寻找 k- 匿名区域，将增加大量的与云服务器的通讯次数，从而引起通讯资费的浪费和速度的限制。

[0027] 2. 该方案也试图利用一次性载入历史数据的下限值 c_{LB} 作为初始数据，然后利用自上而下的算法找到基于历史数据的目标匿名区域作为实际匿名区域计算的初始递归区域，意图来改善初始区域。然而基于历史数据的方法依然采用 Top-Down 的递归方式，而由于根节点所代表的网格区域远远大于实际需求的 k- 匿名目标区域，所以较原始的 Casper

匿名算法的 Down-Top 递归方式，还是大大的增加了递归次数，影响效率。

[0028] 3. 最重要的是，当从云端服务器获取的历史记录得出的移动用户数的下限值出现错误时，如基于历史数据得出的初始位置网格单元的 $c_{LB}> = k$，但是当时的实际移动用户数却刚好小于 k，那么 k- 匿名将失效，移动用户将被暴露，从而导致隐私泄露。所以，此方案的改进又回到了用安全来换取速度和通讯资费的问题，反而违背了隐私保护的初衷。

[0029] 隐私威胁场景建模 ：

[0030] 移动用户 A 发出查询最近餐厅的请求。手机移动终端负责产生相应的匿名区域来保护自己的真实位置。首先从云端一次性载入网格结构每个单元的历史数据，即每个单元的历史移动用户数的下限值。然后得到基于历史数据的匿名区域所在的网格单元 c_{LB}，其 $c_{LB}> = k$，作为实际空间匿名区域算法的初始单元。手机设备将向云端发起请求，查询这个起始区域的当前实际移动用户数。但此时初始单元的实时用户数刚好低于历史最低值并且小于 k，那么，初始网格单元的用户数将不满足 k- 匿名的要求。此时，云端服务器就会知道用户 A 所在区域的移动用户数小于 k，且用户 A 返回的匿名区域中用户数也将小于 k，则 k- 匿名失效，用户 A 的位置隐私泄露，即对 A 的隐私保护失效。

发明内容

[0031] 为了解决上述问题，平衡位置隐私保护、通讯资费和计算速度三者之间的关系，本发明提供了一种适用于移动端位置隐私保护的空间匿名化方法。本发明对空间匿名化方法的改进是为了使得云环境中的移动用户的位置隐私得到很好保护，同时提高计算速度并减少与云服务的交互带来的通讯资费。具体的说，本发明主要是确定算法起始递归区域的可行方法，以避免从根节点递归与云端服务器的多次交互而带来通讯资费的浪费 ；增加相应的安全方案，抵抗现有方法中存在的隐私泄露威胁，实现位置隐私保护。

[0032] 本发明针对如何确定改进的匿名算法的初始递归区域，提出了可行的论证方案，并根据这个过程中可能带来的用户位置隐私泄露的威胁，提出了增加干扰值 m 的方案，从而避免了已有方法中用安全来换取速度和通讯成本的问题，做到了平衡隐私保护、实际速度及通讯资费之间的关系。既保证了隐私保护的有效性又降低了通讯成本。

[0033] 本发明主要内容如下 ：一是，结合 Casper 方法和历史数据确定了自上而下的空间匿名算法的初始递归区域，减少了与云服务器交互的通讯开销 ；二是，提出了两种增加干扰值 m 的方案，适当扩大初始区域，抵抗了现有方法中存在的隐私泄露风险 ；三是，提出了根据实际交通状况进行时间分段的方案，对历史数据进行处理，提高了历史下限值的计算速度和结果的准确性。通过采用 Everyware 实验室提供的 Milano 数据集，在 PC 上模拟了移动用户和云服务器交互生成匿名区域的仿真实验，对改进的方案与现有方案进行性能对比测试，并通过实验给出了干扰值 m 的取值范围。结果表明，本方法以较低的通讯成本和较快的速度为移动用户提供了有效的位置隐私保护。

[0034] 为了使得移动端位置隐私保护的空间匿名化方法继承传统的 Casper 匿名方法的优良性能，即找到尽可能小的匿名区域，与此同时，考虑到云环境中云服务的不完全可信性，我们需要改变 Casper 原始匿名算法的自下而上 (Down-Top) 的递归算法，变为从较大的区域单元向目标区域逐渐缩小的自上而下 (Top-Down) 的递归算法。然而，从较大的区域开始递归虽能产生有效的匿名区域，很好的保证移动用户的位置隐私，但是，这个初始化的区

域是整个系统区域，它远远大于目标匿名区域的大小，从而找到目标匿名区域之前的多次递归，增加了与云服务交互的次数，带来了额外的移动通讯资费和速度的降低。因此，我们针对如何确定改进的匿名算法的初始递归区域，提出了可行的论证方案，并根据这个过程中可能带来的用户位置隐私泄露的威胁，提出了增加干扰值 m 的方案，从而避免了已有方法中用安全来换取速度和通讯成本的问题，做到了平衡隐私保护、实际速度及通讯资费之间的关系。

[0035] 本发明的技术方案为：

[0036] 一种适用于移动端位置隐私保护的空间匿名化方法，其步骤为：

[0037] 1) 移动端采用金字塔型网格数据结构维护空间区域移动用户信息；当进行基于位置的服务查询时，该移动端首先向云服务器查询该空间区域中每个网格单元的移动用户数的历史下限值 c_{LB}；

[0038] 2) 该移动端根据返回的所有网格单元的移动用户数的历史下限值 c_{LB}，对该空间区域进行自下而上的递归，计算出用于实际空间匿名区域计算的初始递归区域；

[0039] 3) 该移动端与云服务器进行交互，从该初始递归区域开始对该空间区域进行自上而下递归询问云服务器，查询网格单元内的当前用户数量，并根据返回的网格单元内的当前用户数量确定是否满足生成匿名区域，如果满足则生成匿名区域，将其发送给基于位置的服务提供方，请求获得相应的服务；

[0040] 4) 该基于位置的服务提供方将查询到的消息队列传返回给该移动端。

[0041] 进一步的，云服务器通过统计该空间区域的移动用户历史数据和用户密度信息，计算出当前该空间区域中所有网格单元的移动用户数的历史下限值 c_{LB}。

[0042] 进一步的，云服务器按不同月份、不同星期、节假日与非节假日和一天中不同时段对统计数据进行分类，根据查询请求所属的时段返回当前网格单元对应的移动用户数的历史下限值 c_{LB}。

[0043] 进一步的，计算出所述初始递归区域的方法为：利用 Casper 匿名算法和每个网格单元的移动用户数的历史下限值 c_{LB}，对该空间区域进行自下而上的递归，直到计算出该空间区域的第 N 层的一网格单元 c^{LB} 包含该移动端的用户，且其移动用户数的历史下限值 c_{LB} 大于或等于该 Cas_per 匿名算法中的设定值 k，将该网格单元 c^{LB} 作为所述初始递归区域。

[0044] 进一步的，设置一干扰值 m，所述设定值 k = k+m, 用于计算基于历史下限值 c_{LB} 的初始递归区域。所述 m 值的取值范围在 3 ～ 5。

[0045] 进一步的，生成所述匿名区域的方法为：用 c^C 表示当前的网格单元，在递归开始时令 $c^C = c^{LB}$；查询 c^C 的四个子网格单元各自的实时用户数，如果其中一子网格单元 c 拥有的用户数不小于设定值 k，则将该子网格单元 c 作为当前单元 c^C，继续进行递归；如果当前层每一子网格单元 c 的实时用户数均小于 k，则检查每一子网格单元 c 及与其垂直相邻网格单元 c^H 的用户总和，或子网格单元 c 及与其水平相邻网格单元 c^V 的用户总和，如果总和不小于 k，并且该子网格单元 c 的邻居网格单元的当前用户数小于 k，则返回该子网格单元 c 和它的邻居网格单元的区域并集作为所述匿名区域，停止递归；否则根据该网格单元 c^{LB} 生成所述匿名区域。

[0046] 进一步的，设置一干扰值 m；从该空间区域的第 N-m 层开始自上而下递归询问云服务器；其中，所述初始递归区域为该空间区域的第 N 层。所述 m 值的取值范围在 0 ～ 3。

[0047] 与现有技术相比，本发明技术方案带来的有益效果

[0048] 随着基于位置的移动服务的普及，移动用户常常会使用基于位置的应用来获得便利。因此，基于位置的移动用户隐私保护变得非常重要。然而，传统的空间匿名化方法需要依赖于单一的第三方可信匿名器，可扩展性差且易于成为攻击的焦点。而设备端的空间匿名化方法，存在速度和通讯成本高的问题，而要避免这个问题则会出现隐私泄露的威胁。本发明针对以上方法存在的问题，结合云服务，提出了一种改进的空间匿名化方法，采用 Everyware 实验室提供的 Milano 数据集，在 PC 上模拟移动用户客户端和云端服务器的交互过程，进行了仿真实验，证明了此方法的有效性与可行性。本发明提出的方法带来的有益效果主要表现在以下方面：

[0049] (1) 针对如何确定改进的空间匿名算法的初始递归位置，提出了一种可行的实现方法，利用云端服务器提供的移动用户历史下限值和 Casper 方法，在手机端自下而上计算，避免了自上而下的递归与云服务器交互带来的不必要的通讯开销。

[0050] (2) 针对构建的隐私泄露威胁场景，提出了两种干扰值 m 方案，并通过仿真实验给出了干扰值 m 方案的取值范围，验证了方案可以很好地抵抗位置隐私威胁。因此，我们改进的方法可以以较低的通讯成本和较快的速度为移动用户提供有效的位置隐私保护。

[0051] (3) 提出了根据实际交通状况把时间分成不同时段的方案，根据不同的时段返回网格单元的移动用户数的历史下限值，从而提高了历史下限值和匿名区域的计算速度和初始递归区域的准确性。

附图说明

[0052] 图 1 为设备端的空间匿名化方法工作流程图；

[0053] 图 2 为适用于移动端位置隐私保护的空间匿名化方法工作流程图；

[0054] 图 3 为适用于移动端位置隐私保护的空间匿名化方法流程图；

[0055] 图 4 为完整金字塔型数据结构；

[0056] 图 5 为空间匿名化方法的不完整型金字塔数据结构；

[0057] 图 6 为 m 值方案一对匿名区域的影响 (m 值不影响匿名区域面积，多条曲线重合)；

[0058] 图 7 为 m 值方案一对匿名时间的影响；

[0059] 图 8 为 m 值方案一对通讯成本的影响；

[0060] 图 9 为 m 值方案二对匿名区域的影响 (m 值不影响匿名区域面积，多条曲线重合)；

[0061] 图 10 为 m 值方案二对匿名时间的影响；

[0062] 图 11 为 m 值方案二对通讯成本的影响；

具体实施方式

[0063] 下面结合附图对本发明进行进一步详细描述。

[0064] 基于网格的完整金字塔型数据结构如图 4 所示，它是分层地把空间区域分解成 H 个层次，其中高为 h 的网格层在水平方向有 4h 个网格单元。金字塔的根节点叫做第 0 层，只有一个网格单元覆盖了整个空间区域。金字塔的每一个单元被表示成 (cid,n)，cid 代表小单元的标识符，n 为此单元边界内的移动用户的数目。金字塔型结构动态跟踪维护了每个小单元中移动用户的数量，使得其保持为当前的实际用户数值。另外，哈希表 (Hash Table)

对于每一个注册移动用户来说都维持一个结构 (uid,profile,cid),其中 uid 代表移动用户标识符,profile 代表用户的隐私属性,cid 代表移动用户所在的单元标识符。cid 通常在金字塔的最低层,如图中的阴影区域。

[0065] 但本发明更需要利用一个不完整的金字塔结构,如图 5 所示。每个网格单元和哈希表的内容与图 4 类似。不完整的金字塔结构的主要思想是只维护那些可以潜在地被用来作为移动用户匿名区域的网格单元。例如,如果所有的移动用户有严格的隐私要求,其中金字塔最底层不能满足任何用户的隐私属性,图 5 的匿名化方法将不维持这一层级,因此,维持金字塔型数据结构的成本显著降低。在图 5 中的阴影网格单元表示被维持的最低层级的网格单元(与图 4 比较可知,图 4 所有的阴影部分仅在最底级)。举例来说,在次高层(即图 5 中的第 1 层)中,右下方的四分之一区域的阴影部分说明,在该区域的所有移动用户都对隐私有严格的要求,任何低级网格单元都不能满足其对隐私的要求。但是,没有必要扩展到整个象限。例如,在最底层,右上角有四个阴影网格单元,这说明这些网格单元的用户对隐私的要求是最宽松的。在空间匿名化方法中,哈希表指向维持网格单元所需的最低的层级,这个层级并不一定是在金字塔的最底层(而图 4 中哈希表指向的是金字塔最底层)。

[0066] 我们以移动用户的整个使用过程来说明改进的空间匿名化方法的工作流程,具体的工作流程如图 2、3 所示。下面详细介绍改进的方法的工作流程。

[0067] 第 1 步:移动用户与云服务器交互。移动设备客户端通过采用不完整金字塔型网格数据结构来维护空间区域移动用户信息 (cid, n)。移动用户请求云服务器,查询每个网格单元的历史移动用户数的下限值(历史最低移动用户数)。

[0068] 第 2 步:云服务器通过统计每个网格单元的移动用户历史数据和区域用户密度信息,计算出当前所有网格单元的移动用户数的历史下限值 c_{LB}。我们可以根据实际交通状况,按不同月份,不同星期,节假日与非节假日,和一天中不同时段进行分类,根据不同的时段返回不同的当前网格单元用户数历史下限 c_{LB},这样可以提高运行速度和结果准确性。

[0069] 第 3 步:移动用户一次性从云端获取所有网格单元的 c_{LB},用来计算基于历史数据的 k- 匿名区域,以用于作为下一步的实际空间匿名区域计算的递归初始递归区域。

[0070] 第 4 步:移动用户在手机设备上负责产生匿名区域。首先,借助 Casper 匿名算法和每个网格单元的 c_{LB},自下而上的递归(即从下面的最底层的包括请求用户的最小网格单元自下而上递归),计算出第 N 层的网格单元 c^{LB},满足:包含当前用户,且其 $c_{LB} > = k$。当然,这个过程中我们增加了干扰值 m,以防止隐私泄露。网格单元 c^{LB} 将作为改进的 Casper 匿名算法的递归初始递归区域,进行后续的实际空间匿名区域的生成。

[0071] 第 5 步:移动用户与云服务器交互。自上而下递归询问云端服务器(即以递归初始单元为起点,向低层递归,逐渐缩小区域),检索在这一时刻某个网格单元内的用户数量。

[0072] 第 6 步:移动用户通过从云端收到的实时用户位置和空间密度信息,独立产生匿名区域。

[0073] 第 7 步:一旦移动用户设备生成一个匿名的区域,就会把空间匿名区域发送给基于位置的服务提供商 (LBS),请求获得相应的服务。如应用“大众点评”,查询最近的 KTV。

[0074] 第 8 步:LBS 会把查询到的消息队列传返回给用户。发起请求的移动用户只要自己筛选出适合自己的信息即可。

[0075] 本发明实施例一:

[0076] 本发明对改进的空间匿名算法的初始递归区域实现方法进行了具体论证。

[0077] 首先，利用云服务提供的每个网格单元的移动用户的历史统计数据，可以得到每个网格单元的历史最低移动用户数量，即移动用户数的历史下限值，记作 c_{LB}，然后一次性的把这些历史下限值下载到手机移动终端，这样就可以本地操作，避免自上而下的每一次递归都与云服务器进行交互带来的通讯开销。当然，这个过程会有一定的初始化成本。又因为每个网格单元的历史数据已经在用户的移动终端上，直接在移动终端计算初始化区域即可，而不需要再与云端交互获取数据，这个过程在本地操作所以不存在不完全可信的云环境问题，因此我们不需要再从最大的区域根节点自上而下递归（前面已经介绍过自上而下的递归会有太多次的递归次数，因为根节点区域远远大于目标区域），而可以采用从最小的网格单元开始自下而上递归的高效 Casper 匿名算法，直接计算出基于历史下限的初始递归区域，这样避免了根节点的区域远远大于目标区域的问题从而提高了速度，降低了通讯成本。所以我们在这里是基于所有网格单元的历史移动用户数的下限值 c_{LB}，在手机终端上借助传统的 Casper 匿名算法，计算出第 N 层的网格单元 c^{LB} 满足：① $_{c}^{LB}$ 区域包含当前用户的位置；② $c_{LB} > = k$。由于历史移动用户数的最低下限代表了历史最低移动用户的人群密度，所以以此算出的匿名区域通常是大于实际匿名区域的。那么，可以将网格单元 c^{LB} 作为下一步计算实际空间匿名区域算法的初始递归区域。

[0078] 一般情况下本方案是安全的，与此同时，本文也考虑到了这样做可能出现的隐私威胁，将于下一小节进行详细分析并提出解决方案，本小节仅用来分析缩小初始递归区域的方法。接下来，我们对确定初始递归区域的算法进行详细分析。

[0079] (1) 基于 c_{LB} 的 Casper 匿名算法

[0080] 我们首先介绍基于 c_{LB} 的 Casper 匿名算法的处理流程。Casper 匿名算法是一种自下而上 (Down-Top) 的递归算法。因为它可以既快又安全的生成尽可能小的匿名区域，所以它在传统的空间匿名架构中能够很好的工作。并且即使攻击者知道匿名化的过程，移动用户的匿名化也可以得到保证。所以我们利用 Casper 匿名算法的思想，借助基于历史数据的每个网格单元的移动用户数的历史下限值 c_{LB}，在移动手机终端执行基于 c_{LB} 的 Casper 匿名算法，从而获得我们想要的下一步实际空间匿名算法的初始递归区域。

[0081] 算法流程：首先从最深层次定位用户所在的相应网格单元 c，然后检查这个特定单元中的历史移动用户数的历史下限值。如果下限值不低于设定的阀值 k，就返回单元 c 作为生成的历史匿名空间；否则，将检查当前的网格单元 c 的水平相邻单元和垂直相邻单元（它们和 c 拥有相同的父节点），分别检查它们和单元 c 的用户历史下限的总和。如果以上两个用户总和数都不小于 k，那么返回总和较大的相邻单元与 c 的并集。如果还不满足要求，则把上一层网格单元即 c 的父节点设为 c，然后重复递归直到网格 G 的根节点。我们通常假设在总区域（与 G 的根节点对应）中的用户数量是远远大于 k 的，因此，此算法将会产生一个包含不少于 k 个用户的区域，从而实现请求所要求的基于历史下限的 k- 匿名区域 R^{LB}，将作为下一步实际空间匿名算法的初始递归区域。

[0082] 具体的基于 c_{LB} 的 Casper 匿名算法：

[0083] 输入：网格结构 G，整数 k，发出请求的移动用户位置 p，每个网格单元的 c_{LB}

[0084] 输出：基于历史最低下限的空间区域 R^{LB}

[0085] 一般 k 通常是远远小于整个区域的人口数量的

[0086] 方法：

[0087]

1：把用户定位在可能达到的最深层数的网格单元 c 中，其中 c 位于第 i 层（取 i 的最大值）

2：**if**(c_{LB}>=k）**then**

3： return 网格单元 c 作为基于历史下限的匿名区域

[0088]

4：**end if**

5：检查 c 的垂直相邻节点 c^H 或水平相邻节点 c^V

6：**if**($c_{LB}^H+c_{LB}$>=k)||($c_{LB}^V+c_{LB}$>=k) **then**

7: **if** c_{LB}^H >=c_{LB}^V **then**

8: return 区域 $R^{LB}=c\cup c^H$

9; **else**

10: return 区域 $R^{LB}=c\cup c^V$

11： **end if**

12: **else**

13: 把 c 的父节点设为 c，然后 repeat 第 2 步

14: **end if**

[0089] (2) 改进的自上而下的匿名算法

[0090] 上一步我们已经通过基于 c_{LB} 的 Casper 匿名算法计算出基于历史移动用户数的下限值的匿名区域 R^{LB}，所在网格单元为 c^{LB}，c^{LB} 所在的层是最接近真实匿名区域所在的层，所以我们利用这个缩小的网格单元作为实际计算匿名区域的初始单元，自上而下递归，逐渐缩小区域直到找到，满足 c_{live}> = k 的最小空间匿名区域，c_{live} 为网格单元的当前实时移动用户数。

[0091] 算法流程：这个算法被设置为从基于历史数据算出的匿名区域 R^{LB} 所在的网格单元 c^{LB} 开始的。用 c^C 表示当前的网格单元，那么在递归开始时 c^C = c^{LB} 首先要查询 c^C 的四个子网格单元 c 各自的实时用户数。请求发起者（移动用户）位于这四个子网格单元 c 中，如果 c 拥有的用户数不小于 k，那么将把 c 作为当前单元 c^C，并重复以上步骤。否则，将检查 $_c$ 和它的垂直相邻节点 c^H 的用户总和或 c 和他的水平相邻节点 c^V 的用户总和。如果总和不小于 k，并且 c 的邻居单元的用户数小于 k，那么返回单元 c 和它的邻居单元的区域并集。与传统的 Casper 匿名算法不同的是，这里还要求 c^H（和 c^V）中的实时用户数量要小于 k。若没有这个要求，如果在 c^H 的实时用户数不小于 k，那么如果用户位于 c^H 中，将直接返回用户数不小于 k 的网格单元 c^H 即可满足要求，没有必要返回两个单元的并集 c ∪ c^H，所以攻击者可以很容易地计算出请求发起者是在单元 c 中而不是 c^H 中，所以此处要求 c^H（和 c^V）中的实时用户数量要小于 k。如果上述子单元 c（并且和它的邻居节点一起）不能满足

k-匿名的要求，算法将返回当前的网格的单元作为匿名区域。当向下递归到达网格最底层，然后停止，把包括请求发起者所在的最底层单元作为返回值（这种情况下，这个单元中的移动用户数不小于 k，因为当前网格单元 c^C 的移动用户数不小于 k，参照下面的算法。所以到达最底层代表 c^C 位于数据结构的最下面，不能再向下递归，所以将返回此时的 c^C 作为匿名区域）。

[0092] 改进的自上而下的匿名算法：

[0093] 输入：网格结构 G，整数 k，移动用户位置 p，网格的最大层数 H

[0094] 输出：生成匿名空间区域 R

[0095] 假设：k 通常是远远小于整个网格区域中的移动用户总数

[0096] 方法：

[0097]

1：把基于历史数据算出的匿名区域 R^{LB} 所在的网格单元 c^{LB} 设置为当前的网格单元，$c^C=c^{LB}$。

2：**if** c^C 不在最深的 H 层 **then**

3：　与云端交互获得 c^C 的四个孩子网格单元的实时用户数

4：　p 位于其中的一个孩子单元 c 中，c_{live} 代表此时的实时移动用数

5：　**if**(c_{live}>=k) **then**

6：　　c^C=c，repeat 第二步

7：　**else**

8：　　**if**($c_{live}+c^H_{live}$>=k) and c^H_{live}<k **then**

9：　　return 区域 R=c∪c^H

10：　　**else if**($c_{live}+c^V_{live}$>=k) and c^V_{live}<k **then**

11：　　return 区域 R=c∪c^V

12：　　**else**

13：　　return c^C /*最差的情况为返回 c^{LB} */

14：　　**end if**

15：　**end if**

16: **else** {到达网格的最底层}

17：　return R= c^C

18：**end if**

[0098] 本发明实施例二

[0099] 本发明增加干扰值 m 应对隐私威胁。

[0100] 上一节分析了如何确定改进的空间匿名算法的初始递归区域，但是把基于历史数据算出的匿名区域 R^{LB} 所在的网格单元 c^{LB} 作为初始网格单元，其实还存在着隐私泄露的危

险。因为当 $c^{LB}> = k$ 时，并不一定保证网格单元 c^{LB} 的实时用户数不小于 k，此时网格单元内的用户数也许刚好是历史上的最低点且小于 k，那么改进的自上而下的匿名算法将返回 c^{LB}，这时，匿名区域的移动用户数量则小于 k，那么 k- 匿名将失效，发起请求的用户位置被暴露。所以，为了更好地平衡速度、通讯成本和位置隐私保护这三个要素，我们对上述方案进行了进一步的改进。

[0101] 我们可以增加一个干扰值适当扩大初始单元的范围，从而提高 k- 匿名的隐私要求，以增加安全性，更好地保护用户位置隐私。我们增加了一个干扰值 m，为了遵循用户友好性原则，发起请求的移动用户在执行程序前可以自行设置 m 的数值，当 m = 0 时，速度最快，但是隐私泄露的风险最高，当 m 为最大值时，隐私保护的效果最好，但通讯成本会因此有一定提高。用户可以根据自己的实际需求选择 m 的值，满足自身的需求。同时，系统也会设定默认值，达到最好的平衡效果，用户可以免去设置的步骤。

[0102] 对于干扰值 m 的增加，本文提出了两种不同的方案。

[0103] 方案一：

[0104] 对于基于 c_{LB} 的 Casper 匿名算法，计算起始单元 c^{LB} 所在的层的位置时，我们用 k+m 代替 k，增大了 k- 匿名的阀值大小，以求获得一个隐私要求更高的更安全的基于历史数据的匿名区域，用来作为改进的自上而下的 Casper 匿名算法的初始网格单元。所以我们修改了基于 c_{LB} 的 Casper 匿名算法的输入部分，用 k+m 替换 k，寻找第一个满足用户数 $c_{LB}> = k+m$ 的网格单元 c^{LB}，作为下一步实际空间匿名算法的初始单元。因为显然有：$(c^{LB}{}_1> = k+m)> = (c^{LB}{}_2> = k)$，即 $c^{LB}{}_1> = c^{LB}{}_2$，从而适当扩大了改进的自上而下的 Casper 匿名算法的初始网格单元的范围，减少了实际隐私泄露的风险。

[0105] 方案二：

[0106] 对于基于 c_{LB} 的 Casper 匿名算法，计算起始单元 c^{LB} 所在的层的位置时，当算法检查到第一个满足 $c_{LB}> = k$ 的网格单元 c^{LB} 时，假设此网格单元位于整个网格的第 N 层，那么我们将把用户所在的单元所在的层向上扩大 M 层，即第 N-M 层（远小于最上层第 0 层），然后把第 N-M 层作为第二步的初始层，设用户在第 N-M 层的四个孩子单元的 c 中，那么单元 c 将作为改进的自上而下的 Casper 匿名算法的初始网格单元。因为 N-M 所在层的区域面积大于 N 所在的面积，所以通过层数的改变适当扩大了基于历史数据算出的匿名区域的范围，增强了 k- 匿名的隐私级别，从而减少了实际隐私泄露的风险。

[0107] 干扰值 m 方案的测试

[0108] 我们设计了两种增加干扰值 m 的方案，分别对这两种方案的 m 值范围进行了测试，实验包括 m 的取值对匿名时间、匿名区域面积和通讯成本的影响，以及增加干扰值 m 后对隐私泄露威胁的抵抗作用。下面分别对两种干扰值 m 方案的测试结果进行分析。

[0109] (1) 干扰值 m 方案一

[0110] 干扰值 m 方案一为使用干扰值 m 变 k- 匿名为 k+m- 匿名，即扩大 k- 匿名的阈值 k 为 k+m，用于计算基于历史下限的初始递归区域。

[0111] 如图 6 所示，此方案在不加 m 值和增加不同数值的 m 值时所生成的匿名区域面积随着 k- 匿名值大小的变化而变化，并且增加不同的 m 值与不加 m 值都返回了相同大小的匿名区域。这说明干扰值 m 方案一对匿名区域的面积没有带来影响，通过了匿名区域指标的测试，可以达到与不加 m 值前一样的性能。

[0112] 如图 7 所示，我们 k- 匿名值取了 2,5,10,15,20 五个值，分别测试了 m 值在 0-20 之间变化对于匿名区域生成时间的影响。可以看出，这五条曲线都是趋于稳定的，说明 m 值方案一中的 m 值在 20 以内对匿名时间的影响不大。

[0113] 如图 8 所示，分别测试了不加 m 值和不同的 m 取值时，通讯成本随 k- 匿名值的变化曲线，我们看出，整体来说 m 值越大，通讯成本越高，但 m ＝ 15 时，通讯成本也没有超过 1.6，说明 m 值对通讯成本的影响不大，并且，当 m 不大于 5 时，通讯成本保值在 1.2 左右或以下，与不加 m 值的通讯成本相似。所以，我们可以把此方案的 m 值范围设定在 0-5 之间。

[0114] 下面我们讨论 m 值方案一对于隐私泄露威胁的抵抗效果。如表 1 所示，1 代表发生隐私泄露，0 代表隐私泄露未发生。可以看出，在未加 m 值时，发生隐私泄露的概率较高，随着 m 值的增加，这种隐私泄露的情况逐渐好转，到 m 不小于 3 时，隐私泄露没有再发生，说明当我们的干扰值 m 取值不小于 3 时，可以很好的抵抗隐私泄露威胁。

[0115] 表 1 使用干扰值 m 变 k- 匿名为 k+m 发生隐私泄露情况对比

[0116]

干扰值 m / k-匿名值	0	1	2	3	4	6	8	10	15	20	30
2	1	0	0	0	0	0	0	0	0	0	0
3	1	1	1	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0
15	1	1	0	0	0	0	0	0	0	0	0
20	1	1	1	0	0	0	0	0	0	0	0
25	1	0	0	0	0	0	0	0	0	0	0
30	1	1	0	0	0	0	0	0	0	0	0

[0117] 综上所述，m 值方案一中，m 值的取值范围在 3-5 之间可以很好的抵抗隐私泄露的威胁，同时保证良好的性能。

[0118] (2) 干扰值 m 方案二

[0119] 干扰值 m 方案二为，使用干扰值 m 减少层数，即用 N-m 代替自上而下计算匿名区域的初始层 N，相当于向上扩大了初始递归区域。

[0120] 如图 9 所示，此方案在 m 值增加前后所生成的匿名区域面积随着 k- 匿名值的大小增大而增大，并且不同的 m 值与不加 m 值都返回了相同大小的匿名区域。这说明干扰值 m 方案二不影响匿名区域的面积，通过了匿名区域指标的测试，可以达到与不加 m 值一样的性能。

[0121] 如图 10 所示，我们 k- 匿名值取了 2,5,10,15,20 五个值，分别测试了 m 值在 0-8 之间的变化，即向上减少 0-8 个层级对于匿名区域生成时间的影响。可以看出，这五条曲线整体时间在 0.08-0.11ms 之间，只有一个点到达了 0.115ms 左右，可以说基本趋于稳定，说明 m 值方案二中的 m 值在 8 以内对匿名时间的影响可以忽略不计。

[0122] 如图 11 所示，分别测试了不加 m 值和不同的 m 取值时，通讯成本随 k- 匿名值变化而变化的曲线，可以看出，九条曲线是趋于稳定的，m 值越大，通讯成本越高，当 m 不大于 3 时，通讯成本不超过 2，所以我们可以把此方案的 m 值范围设定在 0-3 之间。

[0123] 下面我们讨论 m 值方案二对于隐私泄露威胁的抵抗效果。如表 2 所示，1 代表发生隐私泄露，0 代表隐私泄露未发生。可以看出，在未加 m 值时，发生隐私泄露的概率较高，随着 m 值的增加，这种隐私泄露的情况逐渐好转，到 m 不小于 2 时，隐私泄露没有再发生，说明当我们的干扰值 m 取值不小于 2 时，即向上扩大两层就可以很好的抵抗隐私泄露威胁。

[0124] 表 2 使用干扰值 m 减少层数发生隐私泄露情况对比

[0125]

减少的层数 m / k-匿名值	0	1	2	3	4	5	6	7	8
2	1	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0
5	1	1	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0
15	1	0	0	0	0	0	0	0	0
20	1	1	0	0	0	0	0	0	0
25	1	0	0	0	0	0	0	0	0

[0126] 综上所述，m 值方案二中，m 值的取值范围在 2-3 之间可以很好的抵抗隐私泄露的威胁，同时保证良好的性能。

[0127] 本发明实施例三

[0128] 本专利用时间分段代替初始时静态一次更新。

[0129] 基于 c_{LB} 的 Casper 匿名算法，我们是根据所有历史数据计算出历史最低移动用户数的下限值，但是，移动手机用户具有流动性，随着时间段的不同实际的交通状况也会有差别，而不同的时间段移动用户密度也将出现很大的不同，如果每次都根据所有历史数据计算历史下限，虽然能保证是最低值，但也一定程度上影响了计算速度和准确性。例如，当前为上下班高峰时间，那么路上的移动用户密度相对较大，此时如果还用所有时段的历史下限计算历史匿名区域作为实际匿名区域的初始区域，那么这个历史匿名区域将远大于实际的需求区域，因为历史下限值代表了历史上最低的用户密度，那么根据这个最低用户密度算出的满足 k- 匿名的区域将远远大于实际的需求区域大小，那么在后续的计算时，将增加递归的次数，影响效率和准确性。

[0130] 所以我们提出了根据实际交通状况把时间分成不同的时段，根据请求时间的不同计算不同时段的历史下限值，按不同月份、不同星期分类，按节假日和非节假日分类，也可以把一天中的不同时段分类，根据不同的时段返回不同的网格单元移动用户数量的历史下限值 c_{LB}，当用户向云端发出数据请求时，将动态的根据当前的时间段来返回所需的时间段的最低用户密度和数量信息，这样可以提高计算的速度和准确性。

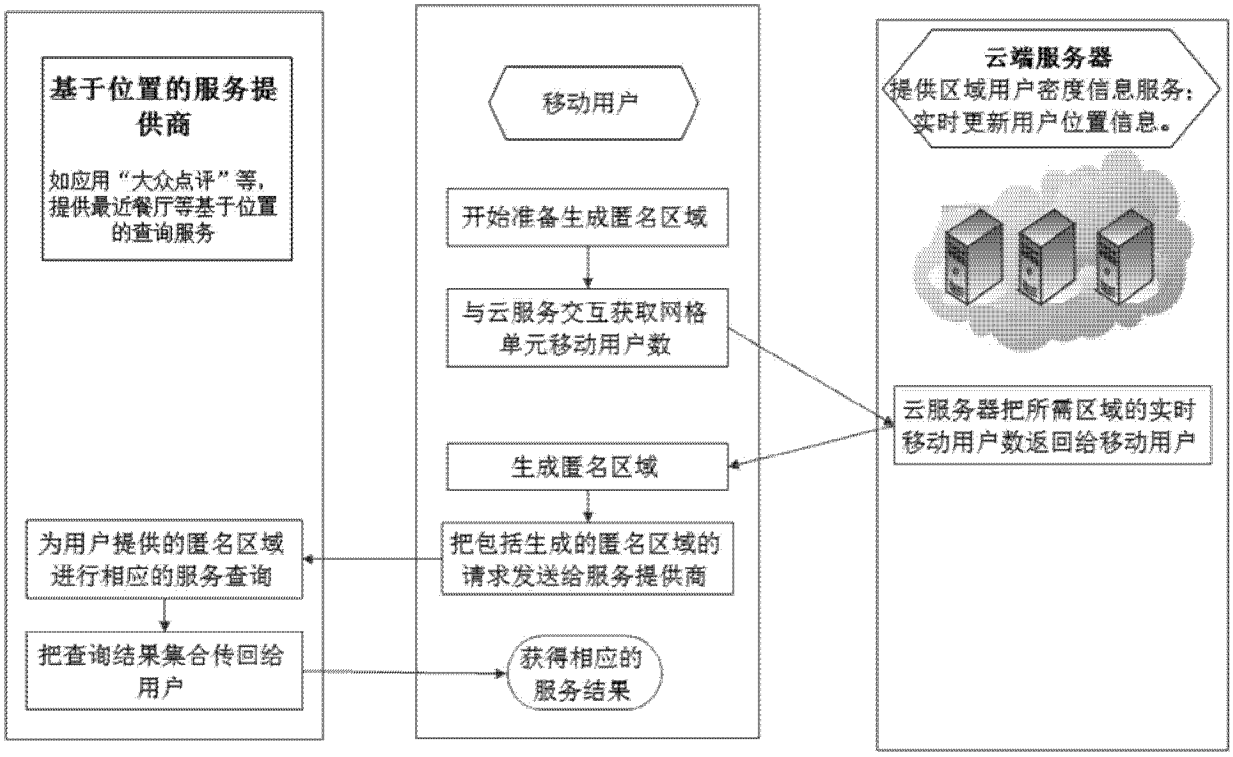

基于位置的服务提供商
如应用“大众点评”等，提供最近餐厅等基于位置的查询服务
移动用户
开始准备生成匿名区域
与云服务交互获取网格单元移动用户数
生成匿名区域
把包括生成的匿名区域的请求发送给服务提供商
获得相应的服务结果
为用户提供的匿名区域进行相应的服务查询
把查询结果集合传回给用户
云端服务器
提供区域用户密度信息服务；实时更新用户位置信息。
云服务器把所需区域的实时移动用户数返回给移动用户

图 1

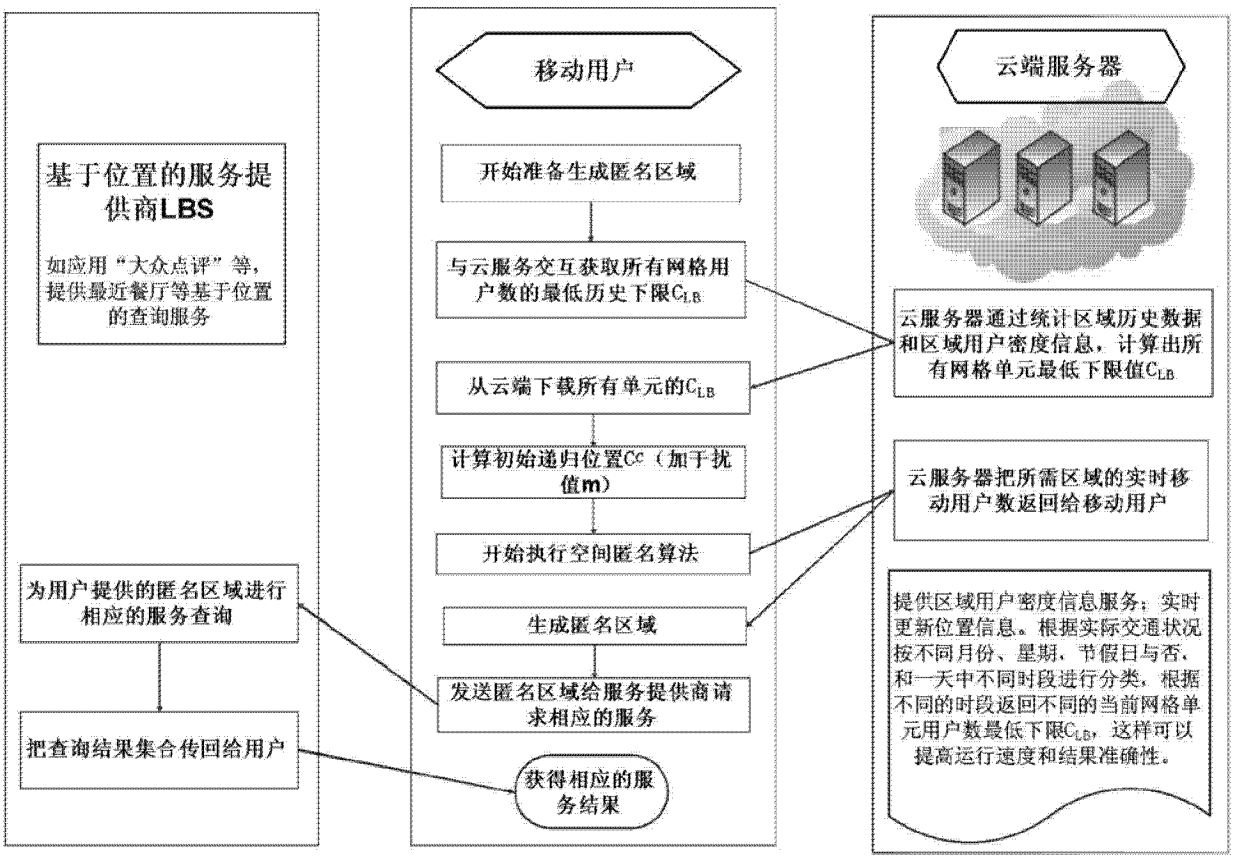

基于位置的服务提供商LBS
如应用“大众点评”等，提供最近餐厅等基于位置的查询服务
为用户提供的匿名区域进行相应的服务查询
把查询结果集合传回给用户
移动用户
开始准备生成匿名区域
与云服务交互获取所有网格用户数的最低历史下限C_{LB}
从云端下载所有单元的C_{LB}
计算初始递归位置Cc（加干扰值m）
开始执行空间匿名算法
生成匿名区域
发送匿名区域给服务提供商请求相应的服务
获得相应的服务结果
云端服务器
云服务器通过统计区域历史数据和区域用户密度信息，计算出所有网格单元最低下限值C_{LB}
云服务器把所需区域的实时移动用户数返回给移动用户
提供区域用户密度信息服务：实时更新位置信息。根据实际交通状况按不同月份、星期、节假日与否，和一天中不同时段进行分类，根据不同的时段返回不同的当前网格单元用户数最低下限C_{LB}，这样可以提高运行速度和结果准确性。

图 2

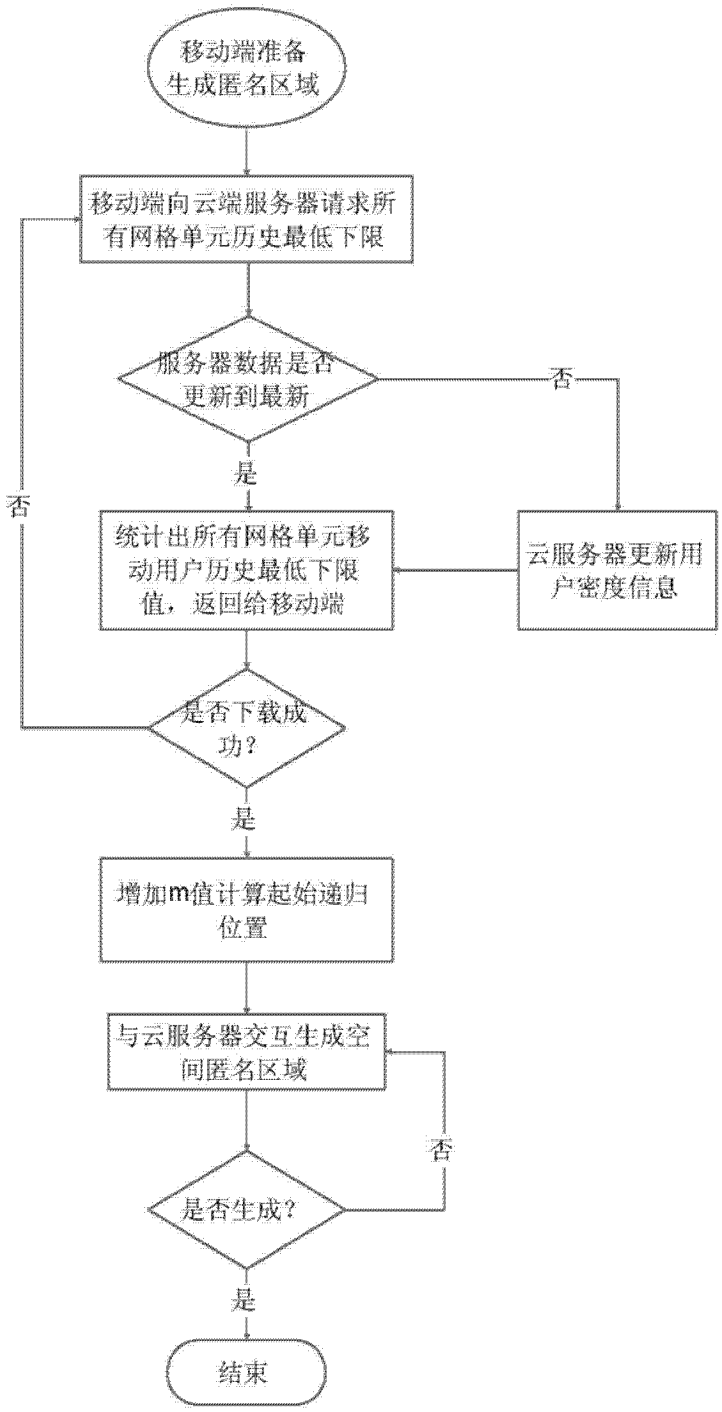

移动端准备生成匿名区域
移动端向云端服务器请求所有网格单元历史最低下限
服务器数据是否更新到最新
否
是
否
统计出所有网格单元移动用户历史最低下限值，返回给移动端
云服务器更新用户密度信息
是否下载成功？
是
增加m值计算起始递归位置
与云服务器交互生成空间匿名区域
否
是否生成？
是
结束

图 3

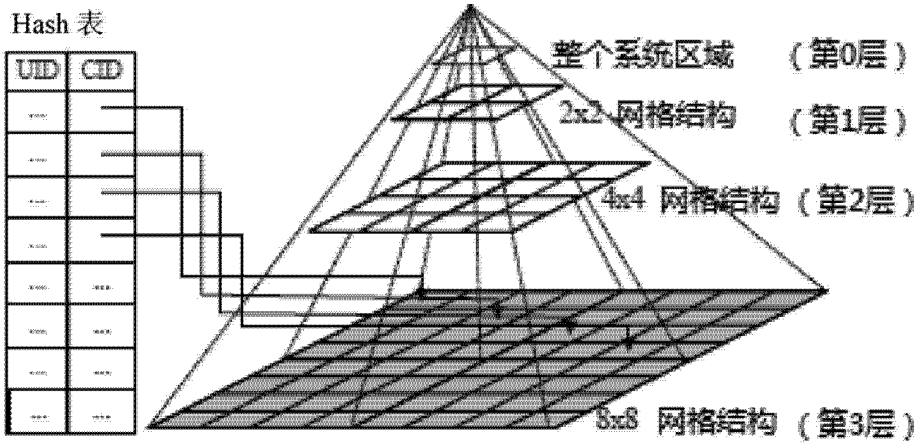

Hash 表
UID
CID
整个系统区域 （第0层）
2x2 网格结构 （第1层）
4x4 网格结构 （第2层）
8x8 网格结构 （第3层）

图 4

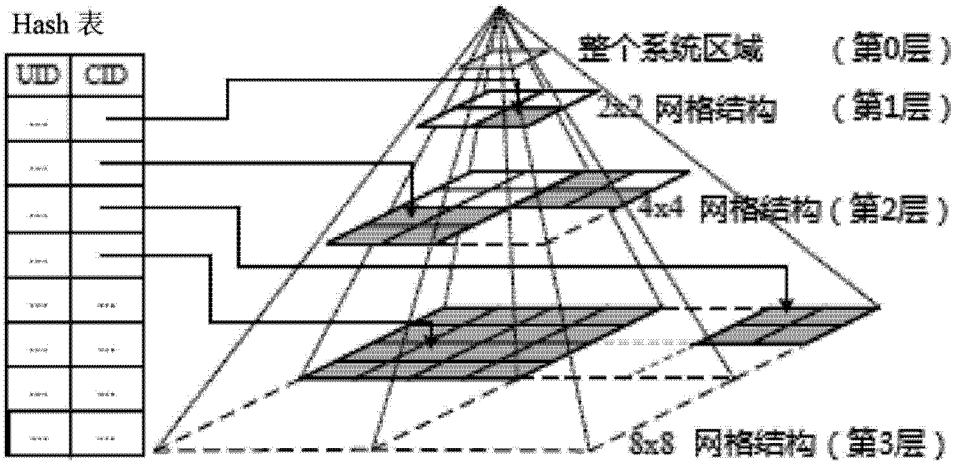

Hash 表
UID
CID
整个系统区域 （第0层）
2x2 网格结构 （第1层）
4x4 网格结构（第2层）
8x8 网格结构（第3层）

图 5

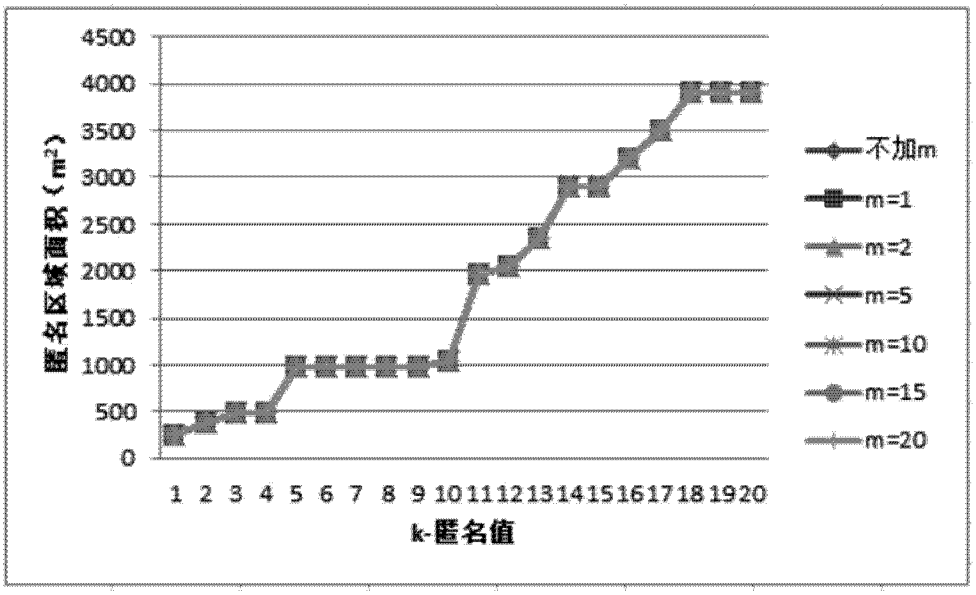

匿名区域面积（m²）
4500
4000
3500
3000
2500
2000
1500
1000
500
0
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
k-匿名值
不加m
m=1
m=2
m=5
m=10
m=15
m=20

图 6

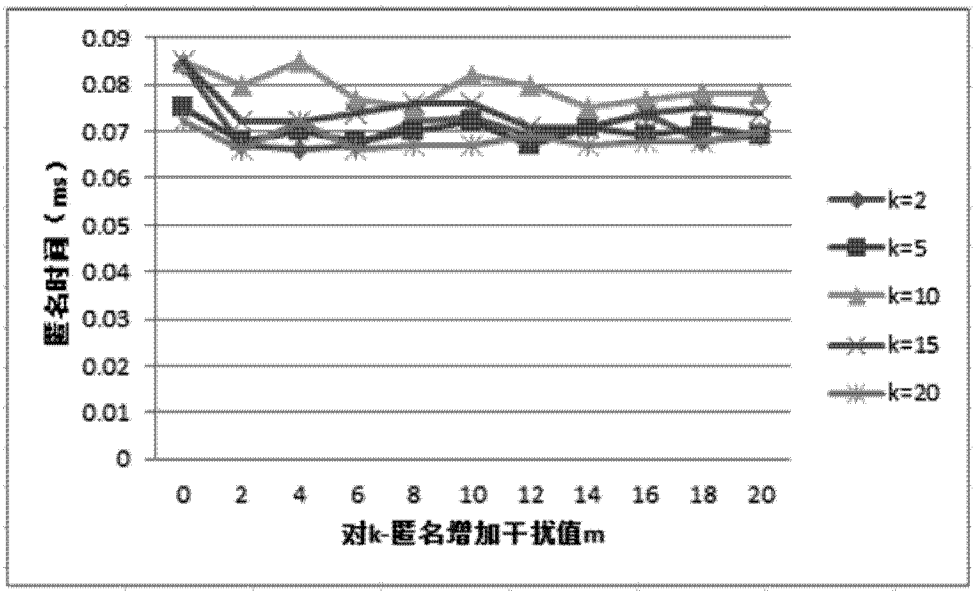

匿名时间（ms）
0.09
0.08
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0
0 2 4 6 8 10 12 14 16 18 20
对k-匿名增加干扰值m
k=2
k=5
k=10
k=15
k=20

图 7

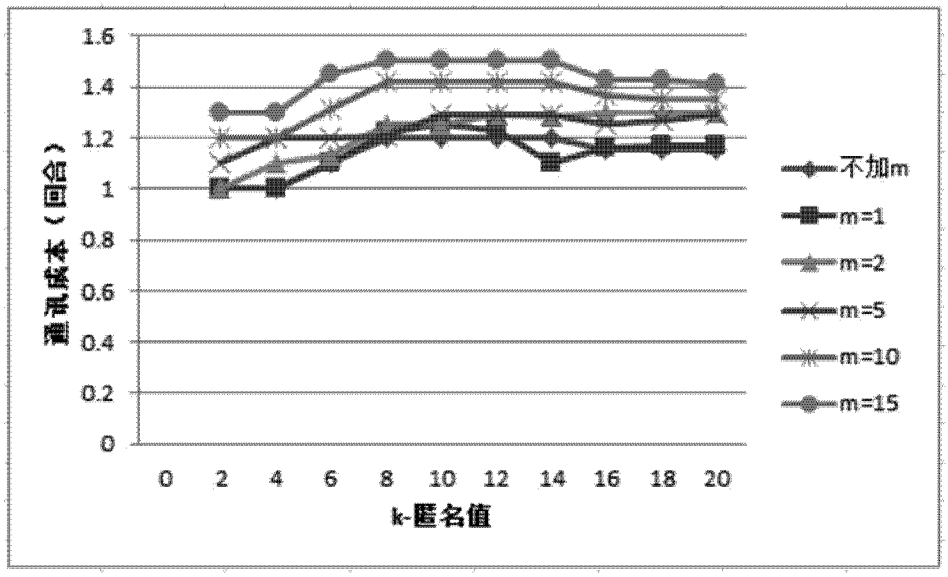

通讯成本（回合）
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
0 2 4 6 8 10 12 14 16 18 20
k-匿名值
不加m
m=1
m=2
m=5
m=10
m=15

图 8

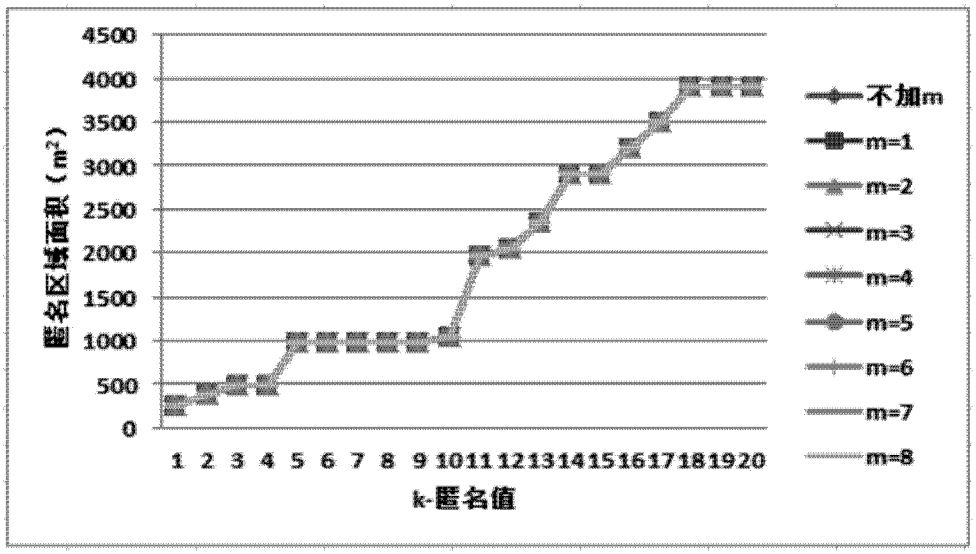

匿名区域面积（m²）
4500
4000
3500
3000
2500
2000
1500
1000
500
0
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
k-匿名值
不加m
m=1
m=2
m=3
m=4
m=5
m=6
m=7
m=8

图 9

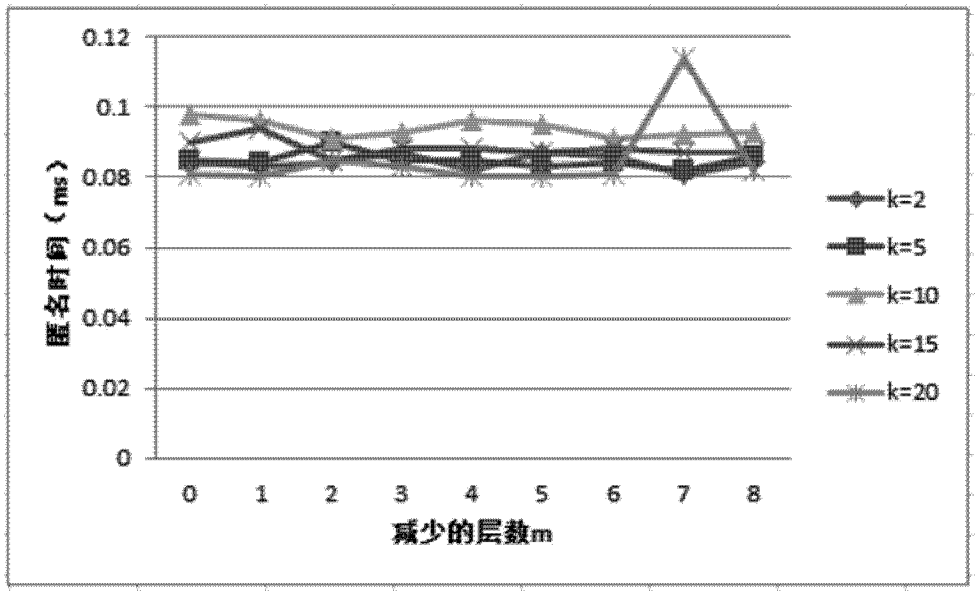

0.12
0.1
0.08
0.06
0.04
0.02
0
匿名时间（ms）
0 1 2 3 4 5 6 7 8
减少的层数m
k=2
k=5
k=10
k=15
k=20

图 10

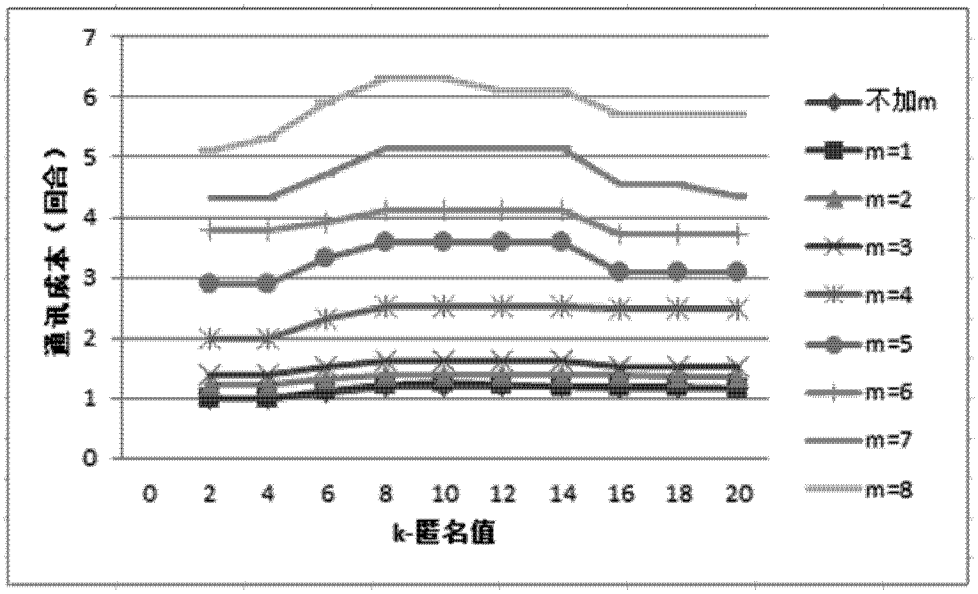

7
6
5
4
3
2
1
0
通讯成本（回合）
0 2 4 6 8 10 12 14 16 18 20
k-匿名值
不加m
m=1
m=2
m=3
m=4
m=5
m=6
m=7
m=8

图 11