面向 Android 应用的细粒度位置隐私保护系统

彭瑞卿[1] 王丽娜[2]
(湖北经济学院信息管理学院 武汉 430205)[1] (武汉大学计算机学院 武汉 430072)[2]

摘 要 位置隐私保护是移动定位服务中的关键安全问题,粗粒度的访问控制机制通过绝对的授权策略抑制了位置信息的暴露,但是忽略了用户的服务质量。提出一种针对本地位置信息的时空模糊算法,实现了细粒度的位置隐私保护系统,在保障用户服务质量的前提下实现位置信息的模糊,从而达到隐私保护的目的。首先设计了一种针对应用程序位置服务请求的位置信息拦截技术,截获精确位置信息,并使用位置模糊算法进行模糊处理;将模糊后的安全位置信息返回给 Apps,从而实现位置隐私保护。实验结果证明了该方法的有效性。

关键词 应用程序,位置服务,服务质量,隐私保护,位置模
中图法分类号 TP309 **文献标识码** A

Fined-grained Location Privacy Protection System for Android Applications

PENG Rui-qing[1] WANG Li-na[2]
(School of Information Management, Hubei University of Economics, Wuhan 430205, China)[1]
(School of Computer, Wuhan University, Wuhan 430072, China)[2]

Abstract The location privacy protection is a prominent problem in mobile location-based services. Coarse-grained permissions methods prevent location data exposure through absolute authorization policies without considering the service qualities of users. In this paper, a native location-based temporal-spatial obfuscation algorithm was proposed to implement the fine-grained location privacy protection system. It can protect user privacy through location cloak, which is also able to keep the service qualities. First, an interception method has been developed to capture the service requests and precise location data sent to applications and processes the location data with location obfuscation algorithm. Then the cloaked safe location data are returned to applications and the location privacy will be protected. The experimental results demonstrate the effectiveness of this method.

Keywords Applications, Location service, Quality of service, Privacy preservation, Location obfuscation

1 引言

随着微型传感器技术的发展,越来越多的移动设备开始装备各种特色智能传感器,位置服务也由单一的导航服务逐步延伸到各种应用服务当中,与移动社交、电子支付等相结合成为当前关注的焦点[1,2]。然而,传感器在智能手机平台上的大量使用在为用户提供越来越丰富的服务的同时,也为用户的隐私安全带来了威胁。例如,通过对 GPS、加速仪、麦克风等传感器数据的统计分析可以推测出用户的位置、移动方式以及社会状态等个人隐私信息[3,4]。

然而位置信息是一种极其重要的个人隐私数据,在某种情况下分享和暴露位置信息的风险要远远超过所获得的服务[5]。应用程序(Applications, Apps)通过采集 GPS 传感器数据,恶意收集大量的超过其正常功能所需的用户位置信息[6],并以此实现非法的信息分享,甚至可以推理用户习惯和社会关系等更高层次的隐私信息,进而导致个人信息泄漏。

如何在不影响用户服务质量的前提下实现细粒度的位置隐私保护是当前的研究重点。访问授权控制是实现隐私保护的一种重要手段[7]。为了控制 Apps 对用户位置信息的访问,许多文献针对服务访问的行为提出了粗粒度的授权策略。例如 Apple 的 iOS 系统平台提供了一个 Location Services 的子系统。Google 的 Android 系统提供了一个类似的子系统,通过选择关闭或开启 GPS 定位功能来实现统一的访问授权策略。然而,当用户选择允许使用位置服务时,则失去了对自身精确位置信息的控制;当选择拒绝时,又无法使用服务。由于这种"允许"或"拒绝"的粗粒度的访问授权机制过于绝对,在限制了权限请求的同时又失去了基于位置的服务,很多情况下并不能达到用户既保护隐私又希望享有位置服务的需求。

本文提出了一种基于位置模糊的细粒度隐私保护方法。当用户处于敏感状态且不能暴露精确位置信息,但是又希望获取位置服务时,通过位置模糊算法,仅暴露模糊的安全位置信息,在保护用户位置隐私安全的同时也保证用户的服务质量。Google 的 Android 系统是目前智能手机使用最为广泛的一种操作系统,本文以 Android 系统为平台,首先拦截 Apps 请求的真实精确位置信息,随后设计了一种针对本地位置信息的时空模糊算法,并通过该算法对截获到的真实位置信息实现位置模糊以达到隐私保护的目的。

2 相关研究现状

前期工作已经考虑到了在使用基于位置的服务过程中,

本文受湖北省教育厅科研计划项目(Q20152203)资助。
彭瑞卿 男,博士,讲师,主要研究方向为信息安全、隐私保护,E-mail:248699762@qq.com(通信作者);王丽娜 女,博士,教授,主要研究方向为信息安全。

用户位置暴露的隐私安全问题[8,9]。其中最为传统的是基于加密的隐私保护方法[8]。Marius Wernke 提出一种基于多密钥共享的位置分享隐私保护方法[9]。通过将精确的位置分割成多个片段，并分布到不同的位置服务器，使各个位置服务器只能获得低精度的位置信息。然而这些加密的方法在不同程度上都需要服务器的参入，在实际的应用过程中，由于用户缺少对服务器的控制，很难根据不同的隐私保护需求对服务器进行频繁的修改，并且无法对方法的有效性进行实际检验。

基于位置模糊隐私保护方法通过隐藏用户的真实位置信息来保护用户的隐私[10]。该方法不需要可信第三方的介入。A. J. Bernheim Brush 通过对用户隐私需求的调查显示了位置模糊对隐私保护重要性[11]；Jayant Venkatanathan 通过对用户的隐私偏好进行分析，控制不同的用户对位置信息的访问程度[12]。但由于这些隐私保护的方法是由应用程序提供的，用户无法控制应用程序对暴露位置信息的使用情况。

为了增强用户对位置使用情况的控制，许多研究者提出了控制应用程序对位置服务的访问授权[15,16]，这些基于移动设备的位置访问控制方法对用户的隐私安全具有重要意义。Fehmi Ben Abdesslem 研究指出默认的同一层次的隐私设置并不能有效地保护用户的隐私，必须要结合上下文的相关信息来合理地选择“允许”或“拒绝”Apps 访问位置信息[15]；Jinseong Jeon 通过将 Apps 的服务请求分割为多个不同类型，并通过修改 Apps 的结构来实现隐私保护[16]；Drew Fisher 对用户的“允许”或“拒绝”Apps 服务请求的授权行为进行建模和分析，以预测用户对 Apps 的决策行为[17]；Hao Peng 提出了一种利用概率生成模型对应用程序的风险进行排序，以协助用户进行“允许”或“拒绝”Apps 服务请求的授权决策[18]。

这些方法在保护用户隐私以及协助用户授权决策方面具有较好的有效性。然而，这种仅提供“允许”或“拒绝”的粗粒度的授权策略无法对位置信息的暴露实现有效控制，当用户处于敏感的时空区域时，依然面临选择即信任的隐私风险。本文基于 Android 系统平台提出了一种基于位置模糊的细粒度隐私保护方法。通过对截获到的本地的位置信息进行模糊处理，实现细粒度的访问授权策略，在保护用户隐私安全的同时，保证了用户的服务质量。

3 细粒度位置隐私保护系统设计

为了在 Android 系统上实现细粒度的位置隐私保护方法，本文利用位置信息拦截技术和访问授权控制技术，设计了一种基于 Android 平台的隐私保护系统，以达到细粒度的位置隐私保护的目的。该系统主要包括位置信息拦截（Location Information Intercept，LII）、细粒度授权控制（Fine-Grained Permissions Control，FGC）以及时空模糊（Temporal-Spatial Obfuscation，TSO）。其系统结构如图 1 所示。

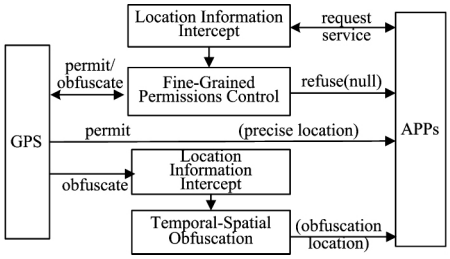


图 1 细粒度位置隐私保护系统结构

下面简要介绍系统的主要功能。

（1）LII。LII 的主要功能是当 Apps 访问 Android 系统中的位置服务时，通过拦截访问过程中的位置信息，获取对位置信息使用的控制。通过对 Apps 的服务请求拦截（request intercept）实现细粒度的访问授权控制，通过对 GPS 数据拦截（GPS intercept）来实现位置模糊处理。

（2）FGC。FGC 的主要功能是通过获得 LII 提供的请求消息，动态地反馈给用户，通过细粒度的授权决策，如：“允许”、“拒绝”和“模糊”，对用户的位置隐私进行细粒度的保护。当用户选择“拒绝”（refuse）时，返回一个空的消息 null 给 Apps，以拒绝访问位置服务；当选择“允许”（permit）时，向 GPS 定位模块发送位置请求，并将精确的位置（precise location）信息发送给 Apps；当选择“模糊”（obfuscate）时，则通过 LII 拦截 GPS 数据，并发送给 TSO，将由 TSO 处理后的模糊位置（Obfuscation location）返回给 Apps。

（3）TSO。TSO 的主要功能是当用户选择“模糊”的授权策略时，对 LII 截获的精确位置数据提供时空模糊算法处理，并将模糊后的安全位置信息发送给 Apps，使得 Apps 可以依据模糊后的位置信息正常地提供服务，并且达到细粒度的隐私保护的目的。

系统的工作流程设计如下。

初始：Apps 发出服务请求；

Step 1 LII 首先拦截到 Apps 发出的位置服务请求，将发出请求的 App 身份标识发送给 FGC 进行处理，转向 Step 2。

Step 2 FGC 在收到身份标识后，对其对应的决策进行判断。若未找到相应决策，则转向 Step 3，否则转向 Step 4。

Step 3 由于未找到 App 对应的决策，则该 App 为首次请求位置服务。向用户发出动态提醒，请求用户对该 App 做出决策，转向 Step 4。

Step 4 若该 App 所对应的决策为“拒绝”，则将空的位置服务回复返回给 App，拒绝其访问到位置服务，本次服务结束。否则，返回正常的位置服务回复给 App，允许其接入到位置服务，并转向 Step 5。

Step 5 若该 App 的服务请求未被拒绝，则可正常接入到位置服务，LII 拦截到位置服务进程返回给该 App 的位置数据信息报文，根据报文中的 App 身份标识找到其对应的决策并进行判断。若为“允许”，则转向 Step 6；否则为“模糊”，转向 Step 7。

Step 6 由于该 App 允许访问精确的位置数据，则将拦截的位置数据正常返回给 App 即可，该 App 完成了一次正常的位置服务。

Step 7 由于该 App 请求的位置数据需要经过模糊处理，因此要通过 TSO 对拦截到的精确位置数据进行模糊，再将模糊后的位置数据返回给该 App，该 App 完成了一次受保护的位置服务。

4 细粒度位置隐私保护系统实现

Android 系统平台对位置服务的访问采用了一个绝对的授权策略，用户通过选择开启或关闭 GPS 定位功能来抑制位置的暴露，无法保障用户的服务质量。本文提出一种细粒度的位置隐私保护系统。通过对位置服务请求拦截技术，获取 Apps 所访问的位置信息，并为用户提供模糊的授权策略，通过时空位置模糊算法，实现细粒度的位置隐私保护。

4.1 位置信息拦截

细粒度访问授权决策实现的前提是必须要拦截 Android

平台中 Apps 对不同层次的位置信息的访问。LII 通过在 Android 平台上的 servicemanage(服务管理)进程和 system_server(系统服务)进程中加载服务请求拦截模块和 GPS 数据拦截模块来实现对位置信息访问的全面控制。

Android 系统中规定,进程之间必须通过 Binder 通信机制进行通信。Apps 要获取位置服务(Location),首先向服务管理进程(servicemanage)发出请求接入 Location 服务的请求数据报。Location 服务在系统服务进程(system_server)中,Apps 根据 Location 服务接入入口向 system_server 发送请求获取用户位置信息的 GPS 请求数据。因此,要加载位置信息截获模块,实现服务请求和 GPS 数据拦截,首先要获取该进程的入口地址。具体过程如下:

首先,查找 elf header(ELF 文件头部)找到 program header(程序头部),program header 包含 ELF 文件中 section(节点)的入口地址。分析发现,system_server 进程的 ioctl 函数偏移地址储存在 rel_plt section 中,然后通过 program header 找到 rel_plt section 的入口地址,将该入口地址赋给指针 rel_plt_table,rel_plt 节头表中每一段内存结构都含有 r_info(函数索引)和 r_offset(对应函数的偏移地址)等信息;根据 ioctl 函数的函数名,实施例首先获取 ioctl 函数的函数索引 name_index,然后和 rel_plt 节头表中的rel_plt_table->r_info(rel_plt 结构中,在指针所指地址内储存的函数索引)进行一一对比,如果 r_info != name_index(不相等),rel_plt_table++(指针指向 rel_plt 的下一个内存结构);如果r_info==name_index(相等),则找到 system_server 进程中的 ioctl 函数偏移地址 io_r_offset=rel_plt_table->r_offset(rel_plt 结构中,在指针所指地址内储存的函数偏移地址),如图 2 所示。

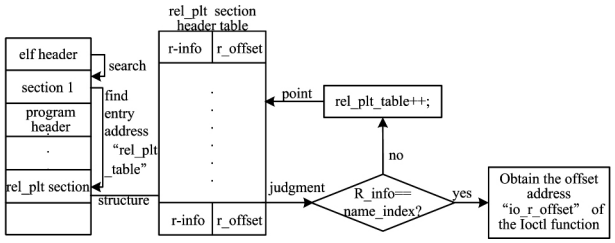


图 2 寻找 ioctl 函数的偏移地址

然后,查看系统的进程列表,查找进程运行的基地址 BaseAddr,根据基地址和 ioctl 函数的入口地址,计算出 system_server 进程和 servicemanage 进程运行时 ioctl 函数的入口地址 ioctl_addr=io_r_offset+BaseAddr。

最后,通过使用 ptrace 函数暂停 servicemanager 进程,并分别在 system_server 进程和 servicemanage 进程中加载和运行 load_server_lib 函数模块,获得该进程的操作权限,使指向 ioctl 函数的指针指向监控模块 hookioctl 函数,从而实现位置信息的拦截。

算法 1 查找 ioctl 函数偏移地址算法

输入:需要查找进程的 ELF 文件格式

输出:所查进程 ioctl 函数的偏移地址 io_r_offset

1. 查找 elf header-> program header;/*找到程序头部*/
2. 查找 program header-> rel_plt section;/*找到 rel_pl 节点*/
3. rel_plt_table=&(rel_plt section)/*赋值给 rel_plt_table*/
4. name_index(ioctl) /*获取索引 name_index*/
5. 将 ioctl 函数索引和 rel_plt section 中的函数信息对比
6. if r_info !=name_index;
7. then rel_plt_table++;/*指向下一个函数*/
8. else then return io_r_offset=rel_plt_table->r_offset;/*返回偏移地址*/

4.2 细粒度授权控制

在对 Apps 对位置信息访问的全面拦截的基础上,本系统为用户提供 3 种不同的细粒度授权策略:允许(permit)、拒绝(refuse)、模糊(obfuscate)。通过对 Apps 访问行为的实时监听提供实时动态的细粒度的授权控制。

首先过滤出带有 Location 服务标签的服务请求数据报。由于 Android 系统中的服务都会在 servicemanager 中进行注册,通过解析服务请求数据报的请求服务字段,查看是否与 Location 服务标签相匹配,若匹配,解析出该服务请求数据报的应用程序 pid(进程号);若不匹配,则不对该服务请求数据报操作。

获取请求位置服务的应用程序 pid 后,通过该应用程序 pid 获得程序名(pidname)写入应用程序行为数据库以方便用户对应用程序行为进行动态的分析。通过对该应用程序行为数据库的分析给出最大化保护用户隐私信息的反馈建议。同时,将该应用程序 pidname 上传给用户,并提供给用户 3 种不同的授权策略:permit,refuse 和 obfuscate。通过选择相应的授权策略,从而实现了细粒度的位置隐私保护。

当面对不可信的 Apps 时,选择“拒绝”,将服务请求的数据包置空,拒绝服务的请求;对于可信的 Apps,选择“允许”,则通过 GPS 定位将精确的位置信息发送给 Apps;当用户处于敏感的环境,希望既不暴露真实的位置信息,又不影响服务的质量时,选择“模糊”,通过对 GPS 数据的截获和控制,并使用基于上下文的时空模糊算法,合理地控制用户的位置使用,并保证服务质量。其实现过程如下。

算法 2 细粒度授权控制算法

输入:截获的进程进程号 pid

输出:位置数据 data

1. pid -> pidname;/*通过进程号查找到进程名所在文件并读取出来*/
2. pidname -> permission;/*将截获进程反馈给用户*/
3. if permission== permit/*允许*/
4. then return data;/*返回原数据*/
5. if permission== refuse/*拒绝*/
6. then return data=null;/*返回空数据*/
7. if permission== obfuscate /*模糊*/
8. then return data=TSC (data,CurrentTime);/*返回经过模糊的数据,模糊算法参见算法 3*/

4.3 时空模糊

为了执行“模糊”的隐私策略,本文设计并实现了一种针对本地位置信息的时空模糊算法。对截获到的真实位置信息进行模糊处理,从而保障服务质量和隐私安全。

设移动用户 u 的移动轨迹形式为:$T=\{(p_1,t_1,v_1),(p_2,t_2,v_2),\cdots,(p_n,t_n,v_n)\}$,其中,$p_i=(x_i,y_i)$是用户 u 在时间 t_i $(1\leqslant i\leqslant n)$时的位置坐标点。$v_i$ 为用户在时间 t_i 的瞬时速度。算法结构如下。

算法 3 TSC

输入:GPS 拦截获取到的精确位置 P 和时间 t;约束时间 δ;

输出:模糊的安全位置 P′。

1. 获取初始位置点 P_0,t_0
2. 对 P_0 执行 SC,计算 P_0';

3. 第 i 次获取的位置点 P_i，t_i；

4. if $|t_i-t_{i-1}|\leqslant\delta$→执行 TSC，计算 P_i'；

5. else→执行 SC，计算 P_i'；

6. 返回 P_i'；

(1)空间模糊 SC。根据最小模糊区域的隐私安全性要求[19]，认为两个轨迹 T，T'之间的最小边界矩阵 $AREA(MBR(\{T,T'\}))>L$ 时，满足隐私要求。

如果 $|X_1-X_1'|>o$ 与 $|Y_1-Y_1'|>o$ 同时成立，则必然存在：

$$AREA(MBR(X_1,Y_1),(X'_1,Y_1'))=|X_1-X_1'||Y_1-Y_1'|>L \quad (1)$$

其中，$[X_1-o,X_1+o]$与$[Y_1-o,Y_1+o]$为点(X_1',Y_1')的禁止取值区域，L 为区域面积。由此得出模糊后的假轨迹点 (X_1',Y_1')的取值范围为：

$$\begin{cases} X_1'^2+Y_1'^2\leqslant R^2, & R>0 \\ X_1'>X_1-o, & X_1'<X_1+o \\ Y_1'>Y_1-o, & Y_1'<Y_1+o \end{cases} \quad (2)$$

(2)时间模糊 TC。为了限制两个相邻时刻的所有位置点之间的距离小于用户以当前移动方式下的最大可能移动速度所能达到的距离。设 Max_v，Min_v 分别为该取值的临界最大、最小值，d 为当前的实际距离，设 $a\cdot v$，$b\cdot v$ 分别为用户速度的可能最大值和最小值。则移动速度的限制条件为：

$$Max\_v=2R+b\cdot v\cdot d=2k\sqrt{L}+b\cdot v\cdot d \quad (3)$$

$$Min\_v=2\sqrt{L}+a\cdot v\cdot d \quad (4)$$

5 实验验证

为了验证提出的细粒度位置隐私保护系统的有效性，本文分别从以下几个方面进行了实验：时空模糊算法效率、位置信息拦截和细粒度授权控制方法的有效性和时间效率。

5.1 实验设计

本次实验所处的硬件环境为小米手机 2，系统为 Android 4.1，CPU 频率为四核 1.5GHz，CPU 型号为高通 APQ8064，GPU 为 Adreno 320，内存为 2GB，网络制式为 WCDMA/GSM，GPS 为 GPS+GLONASS AGPS。任意选取一个基于位置服务的 App 作为保护对象，如百度地图。

首先，验证了位置信息拦截和细粒度授权的有效性。通过对一定数量的 Apps 的拦截和授权分析来体现细粒度授权的有效性。其次，验证了模糊算法的安全性。检验了模糊算法是否满足隐私安全性要求。然后，验证了使用本文提出的位置隐私保护系统对服务质量的影响。检验使用本文提出的细粒度位置隐私保护方法之后，该 App 是否可以正常提供服务。最后，验证了整体系统的时间效率。由于 Android 平台下应用程序的效率主要表现为时间的消耗。通过对比使用本文的隐私保护和正常未加载时获取位置信息的时间消耗，验证系统的效率。

5.2 细粒度授权控制分析

本部分实验对目前 Google Apps 平台上最受欢迎的 66 个 Apps 的使用情况以及应用服务类型进行了分析。分析发现约 24%的 Apps 需要精确的位置信息才可以提供服务，约 63%的 Apps 只需要模糊的或假的位置信息，同样可以提供服务，约 13%的 Apps 的服务完全不需要位置信息。其中比较有代表性的位置服务 Apps 分布情况如图 3 所示。

图 3 不同 Apps 授权策略比较

实验结果显示，本文所提出的服务请求拦截可以全面地拦截 Apps 对位置服务的请求，并根据用户的偏好进行细粒度的授权。

5.3 算法安全性分析

为了验证时空模糊算法的有效性，在不同的移动速度下，通过计算相邻时刻任意两位置点之间的最大距离来分析其有效性。

将 $v\cdot|T_a-T_b|$ 的值分别取 10，15，20，25，30，a，b 分别为 0.5 和 1.5。在取每一个值的情况下，分别取 20 次数据，取出下一个时间节点产生的假轨迹点到前一时间节点的所有位置点的最大距离 D_max 与 $v\cdot|T_a-T_b|$ 的对应关系，如表 1 所列。

表 1 相邻时刻位置点之间的距离比较

$v\cdot\|T_a-T_b\|$	10	15	20	25	30
D_max	9.3414	13.9887	19.0536	24.4408	28.5769

通过实验发现所产生的假轨迹点完全符合要求，可以有效地防止移动模型的攻击。

为了验证算法隐私安全性，定义轨迹所在区域为 AS，区域内的每个网格为一个 $Grid$。通过每个 $Grid$ 内的拥挤度，即 $Grid$ 内位置点数量来体现隐私保护程度。本文在取 $Grid$ 的值为 $8*8$，移动速度 v 为 20m/s，轨迹隐私保护度 k 为 5 时，对算法的隐私保护度进行测试，结果如图 4 所示。

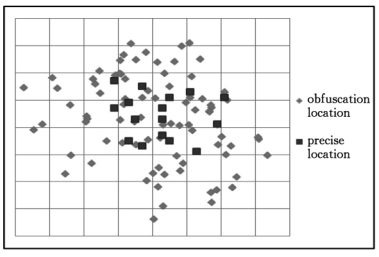


图 4 时空模糊算法隐私性度量

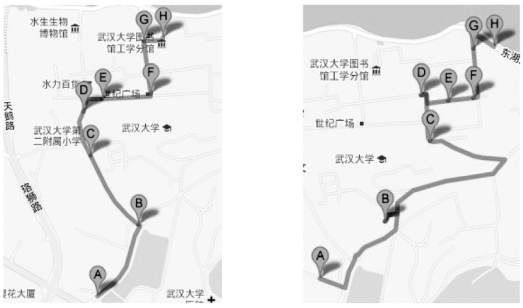
如图 4 所示，在每个拥有真轨迹点的 $Gird$ 中，都几乎有 4 个假轨迹点的存在，真轨迹的隐私保护度达到 5，即满足 k 匿名，达到隐私保护的需求。

5.4 服务质量分析

在移动手机上安装并运行本文提出的算法，采集用户真实的移动轨迹，并记录通过时空模糊算法所生成的假轨迹。截取部分轨迹片段，并将记录的位置数据在地图中展示，如图 5 所示。

实验结果显示，系统可以正常使用位置模糊算法对真实

的位置点进行修改，并根据反馈的模糊位置数据正常提供定位服务。因此，该方法保证了用户的服务质量。



(a)真轨迹　　(b)假轨迹

图 5

5.5 系统效率分析

本节实验通过分阶段加载拦截模块以及时空模糊算法，验证本文提出的隐私保护系统的效率。由于 Android 平台下应用程序的效率主要表现为时间消耗，并且 Apps 的效率与智能手机型号和系统版本无关，因此，本节实验将在同一个手机和系统环境下进行时间消耗的测试。对比使用本文的位置模糊隐私保护处理的时间消耗 T_1 和正常允许的时间消耗 T_2，记录 App 从开始运行到该 App 收到反馈消息为止的时间消耗。由于应用程序在获得反馈消息之后的处理时间只与 Android 平台的性能相关，因此该段时间不作为评价标准。将每一个加载阶段视为一组实验，在不同阶段分别运行 20 次取平均时间消耗。如表 2 所列，T_1 为隐私保护消耗时间，T_2 为正常运行消耗时间，时间的单位为毫秒。

表 2　不同运行模式下的平均时间消耗比较

T_1	T_2	rate of rise
1001.222	1010.4	$9.2*10^{-3}$

实验结果显示，本文提出的隐私保护系统具有较好的效率，并且在隐私保护过程中不会给用户带来过多的消耗。

结束语　本文基于 Android 平台提出了一种基于位置模糊的细粒度位置隐私保护系统。该系统的实现为用户的位置信息提供细粒度的隐私保护。当用户处于敏感状态不能暴露精确位置信息，但是又希望享受位置服务时，通过对截获到的真实位置数据进行模糊处理，并将处理后的安全位置信息发送给 Apps，实现了细粒度的隐私保护。该系统在保护了用户的隐私安全的同时不影响用户的服务质量。实验结果显示，所提方法在服务质量、时间效率、隐私保护等方面都具有较好的效果，并且该方法的提出为隐私保护算法在 Android 平台的实际应用提供了技术保障。在后续工作中，将基于该技术平台，综合考虑更为丰富的上下文信息对隐私保护的影响，进一步研究更为安全有效的隐私保护技术。

参 考 文 献

[1] 李建中，高宏. 无线传感器网络的研究进展[J]. 计算机研究与发展，2008，45(1)：1-15

[2] 刘经南，郭迟，彭瑞卿. 移动互联网时代的位置服务[J]. 中国计算机学会通信，2011，7(12)：40-50

[3] 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

[4] Zhu Y，Sun Y，Wang Y. Nokia Mobile Data Challenge：Predicting Semantic Place and Next Place via Mobile Data[C]// Mobile Data Challenge by Nokia. 2012：1-6

[5] Tsai J Y，Kelley P G，Cranor L F，et al. Location-Sharing Technologies：Privacy Risks and Controls[J]. I/S：A Journal of Law and Policy for the Information Societ，2010，6(2)：1-26

[6] Enck W，Gilbert P，Chun B G，et al. TaintDroid：an information-flow tracking system for realtime privacy monitoring on smartphones[C]// the 9th USENIX conference on Operating systems design and implementation (OSDI'10). Berkeley，CA，USA：USENIX Association，2010：1-6

[7] Kelley P G，Benisch M，Cranor L F，et al. When are users comfortable sharing locations with advertisers?[C]// The 2011 Annual Conference on Human Factors in Computing Systems (CHI'11). New York，USA：ACM Press，2011：2449-2452

[8] Qianhong W，Bo Q，Zhang L. Bridging Broadcast Encryption and Group Key[C]// International Association for Cryptologic Research 2011. 2011：143-160

[9] Wernke M，Durr F，Rothermel K. PShare：Position sharing for location privacy based on multi-secret sharing[C]// Pervasive Computing and Communications. Lugano：2012：153-161

[10] Duckham M，Kulik L. A Formal Model of Obfuscation and Negotiation for Location Privacy[C]// Pervasive Computing. Lecture Notes in Computer Science. 2005：152-170

[11] Brusha J B，Krumm J，Scott J. Exploring end user preferences for location obfuscation，location-based services，and the value of location[C]// The 12th ACM International Conference on Ubiquitous Computing (Ubicomp'10). New York，USA：ACM Press，2010：95-104

[12] Venkatanathan J，Lin J，Benisch M. Who，when，where：Obfuscation preferences in location-sharing applications[R]. CMU-CyLab-11-013，2011：1-12

[13] Au K W Y，Zhou Y F，Huang Z，et al. PScout：analyzing the Android permission specification[C]// The 2012 ACM Conference on Computer and Communications Security (CCS'12). New York，USA：ACM Press，2012：217-228

[14] Barrera D，Clark J，McCarney D，et al. Understanding and improving app installation security mechanisms through empirical analysis of android[C]// The Second ACM Workshop on Security and Privacy in Smartphones and Mobile Devices(SPSM'12). New York，USA：ACM Press，2012：81-92

[15] Ben A F，Henderson T，Brostoff S，et al. Context-based Personalised Settings for Mobile Location Sharing[C]// Proceedings of the ACM Recommender Systems Workshop on Personalization in Mobile Applications. 2011

[16] Jeon J，Micinski K K，Vaughan J A，et al. Dr. Android and Mr. Hide：Fine-grained Permissions in Android Applications Categories and Subject Descriptors[C]// SPSM'12 The Second ACM Workshop on Security and Privacy in Smartphones and Mobile Devices. 2012

[17] Fisher D，Dorner L，Wagner D. Location Privacy：User Behavior in the Field Categories and Subject Descriptors[C]// The Se-cond ACM Workshop on Security and Privacy in Smartphones and Mobile Devices (SPSM'12). North Carolina，USA：2012：51-56

[18] Peng H，Gates C，Sarma B. Using Probabilistic Generative Models for Ranking Risks of Android Apps[C]// The 2012 ACM Conference on Computer and Communications Security (CCS'12). New York，USA：2012：241-252

[19] Ghinita G，Kalnis P，Khoshgozaran A，et al. Private queries in location based services：anonymizers are not necessary[C]// the 2008 ACM SIGMOD International Conference on Management of Data. 2008：121-132