面向移动社交网络的数据共享与访问控制*

张琳,冯青,郭耀+,陈向群

(高可信软件技术教育部重点实验室，北京大学信息科学技术学院软件所，北京 100871)

摘 要 目前，大部分社交网络采用的访问控制机制只能在较粗的粒度上划分数据的级别，并且不支持用户与不同的联系人分享数据的不同部分. 提出了一种基于社交关系度的数据共享与访问控制机制，使数据拥有者可以为亲疏有别的用户分配不同的访问权限,从而可以灵活地控制数据(或部分数据)的流向，更好地保护数据安全和个人隐私。在 Apache 服务器和 Android G2 手机上建立了系统的原型，并以手机通讯录作为数据共享的一个实例进行了验证。

关键词 移动社交网络，访问控制，社交关系度，有向社交关系图

中图法分类号 TP302 **文献标识码** A

Data Sharing and Access Control for Mobile Social Networks

ZHANG Lin, FENG Qing, GUO Yao+, CHEN Xiang-Qun

(Key Laboratory of High Confidence Software Technologies (Ministry of Education)
Institute of Software, School of Electronics Engineering and Computer Science, Peking University, Beijing 100871, China)
+ Corresponding author: Phn: +8610-62753496, Fax: +8610-62751792, E-mail: yaoguo@sei.pku.edu.cn

Abstract Current access control policies on most online social networks only provide coarse-grained data classification. Users can not share different parts of data with different contacts flexibly. This paper introduces a data sharing and access control mechanism based on social degrees, so that data owners can delegate different rights to different users to provide fine-grained data security and privacy. In order to demonstrate the feasibility of the proposed mechanism, we implemented a prototype system on an Apache server and Android G2 smart phones, using contacts sharing as a case study.

Keywords mobile social networks, access control, social degree, directed social graph

* Supported by the National Basic Research Program of China (973) under Grant No. 2009CB320703（国家重点基础研究发展规划(973)）; the Science Fund for Creative Research Groups of China under Grant No. 60821003(国家创新研究群体); Key Science & Technology Special Projects under Grant No. 2009ZX01039-001-001(国家科技重大专项）; the National High-Tech Research and Development Plan of China under Grant No. 2007AA010304, 2009AA01Z139－1（国家高科技研究发展计划).

1 引言

作为便捷的数据共享平台，社交网络在为用户提供数据共享服务的同时，还引入了访问控制机制以保障数据的安全。其核心思想是由用户来标注数据的等级(如隐私(private)、公开(public)、好友可见(accessible by direct contacts)等等)。例如，Bebo、Facebook 和 Multiply 的用户可以与选定的好友分享数据；Friendster 和 Orkut 的用户可以与“好友的好友(friends of friends，即二度好友(second-degree friends))”共享数据；Xing 的用户甚至可以在三度(third-degree friends)、四度好友(fourth-degree friends)之间彼此分享数据。

然而大多数情况下，由于这样等级划分的粒度较粗，在具体应用时或是太严格、或是太笼统，并不能恰到好处地符合用户的需求。例如，在人人网上分享相册，用户仅能在如下五个级别中选择：自己，密码访问，我的好友，好友及同城同公司同学校的人或者所有人。同时，对于用户分享的数据，系统的其他用户或是取得全部，或是一无所获(all-or-nothing)。数据的拥有者不能设置更细粒度的策略使得数据对某些用户部分可见。

随着智能手机的兴起，大型社交网站(如 MySpace, Facebook 等)都开始转向移动领域，一些专门的移动社交网络(如 Foursquare, Gowalla 等)也相继出现。可以预见，由一部部手机搭建起来的超大规模移动社交网络的形成已是大势所趋。

本文提出了一套基于社交关系度的数据共享与访问控制机制，使得用户能够更清晰的阐述自身的共享需求、细粒度地划分数据的等级，从而可以灵活地控制数据(或部分数据)的流向、更好地保障数据安全和个人隐私。我们在 Android G2 手机(结合 Apache 服务器)上实现了上述机制，并验证了该机制的有效性。本研究的最终目标是期望为手机用户(包括上层应用程序)提供一种安全、透明的通用数据共享方式。

本文的组织结构如下：第二部分阐述了基于社交关系度的数据共享与访问控制机制的核心思想；第三部分介绍了平台的设计；第四部分描述了平台的实现，并以智能手机上通讯录的共享作为数据共享的一个实例；第五部分对相关工作进行了简要的介绍；第六部分总结了本文的工作以及对未来工作的展望。

2 基于社交关系度的数据共享与访问控制

现实世界中存在着复杂的社会关系，反映了人与人之间的亲疏远近：如家人、朋友、同学、同事以及商业伙伴等等。本文提出以社交关系度的概念来代表人与人之间的亲密程度，以及一种基于社交关系度的数据共享与访问控制机制，使用户可以对社交关系度不同的联系人设置细粒度的访问权限，从而可以更好地保护数据安全与个人隐私。

2.1 有向社交关系图

为了更好地表示社交网络中用户之间的关系，本文提出了**社交关系图**(directed social graph)的概念，其中以用户为结点，以社交关系为边。考虑到现实中往往存在着不对等的社交关系，例如，用户 A 视用户 B 为密友，但是用户 B 仅将用户 A 看作普通朋友。所以我们采用带权重的有向边来表示社交关系，从而构成**有向社交关系图**。

设存在用户 U_i 和 U_j，连接两个用户的有向边 E_{ij} 和 E_{ji}。则 E_{ij} = <U_i, U_j, *RelationType*>表示用户 U_j 视用户 U_i 为 *RelationType* 类型的朋友；而 E_{ji} = <U_j, U_i, *RelationType*>则表示用户 U_i 视用户 U_j 为 *RelationType* 类型的朋友。

以 *CloseFriends* 代表密友关系，以 *Friends* 代表普通朋友关系。那么，在上面的例子中，用户 A 和 B 之间的社交关系可以表述为：

E_{AB} = <U_A, U_B, *Friends*>

E_{BA} = <U_B, U_A, *CloseFriends*>

图 1 是一个有向社交关系图的实例。

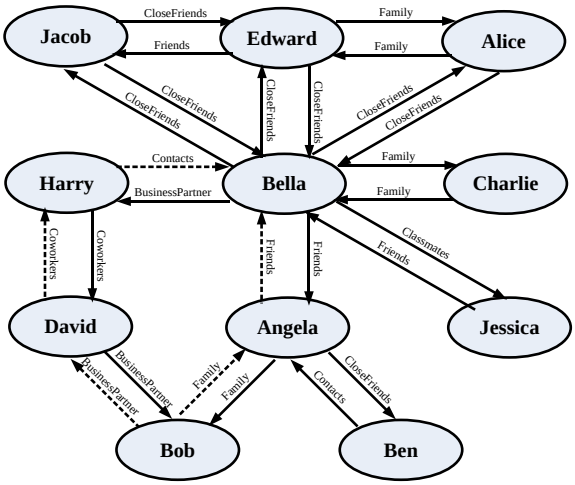

Figure1 A Directed Social Graph
图 1 有向社交关系图

2.2 社交关系度

为了衡量两个用户之间的亲疏程度，本文提出了**社交关系度**(social degree)的概念。社交关系度用数字形式量化地表示了用户之间的亲密程度，其数值越小，说明关系越紧密。举例来说，我们可以采用如下的社交关系度模型：对任一用户而言，自身的社交关系度为 0；所有直接联系人的社交关系度为 1；所有间接联系人的社交关系度为 2；依次类推，根据“六度分割理论[1]”，我们设最远的社交关系度为 6。

另外，在 0～1 之间，用户可以根据自身的情况，为不同的社交关系分配合适的社交关系度。例如，推荐的默认设置为，家人(Family, 0.1)，密友(Close Friends, 0.2)，朋友(Friends, 0.3)，同学(Classmates, 0.4)，同事(Coworkers, 0.5)和商务伙伴(Business Partners, 0.6)。用户可以对上述默认设置进行增添和调整。

根据 2.1 小节所述的有向社交关系图，社交关系度可以进行叠加计算。我们定义 P_{ji} 表示从用户 U_j 出发，沿社交关系有向边到达用户 U_i 的一条有向无环通路；p_{ji} 表示所有从用户 U_j 出发，沿社交关系有向边到达用户 U_i 的有向无环通路中，经过结点数最少的一条；则对用户 U_i 而言，用户 U_j 的社交关系度为路径 p_{ji} 上的结点数（U_i, U_j 除外）与该路径上从 U_j 出发的社交关系有向边上的社交关系度之和。

例如，如图 1 所示，计算对 Bella 而言，David 的社交关系度。图中以虚线标出了两条从 David 出发沿社交关系有向边到达 Bella 的有向无环通路。其中，一条通路只经过一个结点 Harry；而另一条通路则经过两个结点：Bob 和 Angela。故以第一条通路为准，计算对 Bella 而言，David 的社交关系度为：1 + 0.5 = 1.5 [2]。

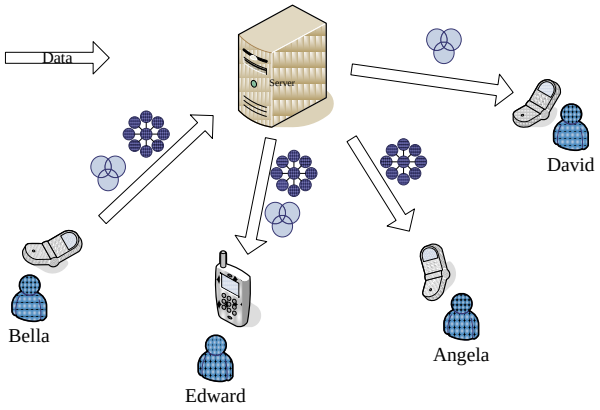

Figure 2 A Data Sharing Example

图 2 数据共享实例

2.3 基于社交关系度的数据共享与访问控制

访问控制的核心思想是拦截每一次对数据(或资源)的访问请求，然后根据一组既定的规则，判断该请求是否应当被授权。这一组既定的规则，就是**访问控制策略**。传统的访问控制策略涉及三类实体：主体、客体和权限。一个由主体 A、客体 O 和权限 R 组成的三元组<A, O, R>表示主体 A 对客体 O 具有权限 R。基于社交关系度的访问控制策略在上述三元组的基础上，对客体 O 进行拆分，并引入第四个分量，社交关系度。一条基于社交关系度的访问控制策略<A, O: O1| O2|…|On, D, R>表示，对主体 A 而言，社交关系度小于(或等于)D 的主体，对客体 O 中的 O1，O2，…，On 具有权限 R。

如图 2 所示，用户 Bella 通过手机共享相册 Rings。并设置访问控制策略<Bella, Rings, CloseFriends, Read>，<Bella, Flowers: nine-rings, Friends, Read> 和 <Bella, Flowers: three-rings, 2, Read>。

则作为密友，用户 Edward 可以获取全部照片；作为普通朋友，Angela 只能获得名为 nine-rings 的照片；而对 Bella 而言，由于 David 的社交关系度为 1.5，故只能浏览名为 three-rings 的照片。

3 平台设计

利用上述基于社交关系度的数据共享与访问控制机制，我们设计了一个面向移动社交网络的数据共享平台。该平台中包含服务器端和手机客户端两个部分，系统架构如图 3 所示:

3.1 服务器端

服务器端分为三个部分：用户管理中心(User Manager)、数据管理中心(Data Manager)和访问控制策略管理中心(Access Control Policy Manager)。

用户管理中心管理用户名[3]和密码，负责验证某一特定用户是否为系统的有效用户。**访问控制策略管理中心**管理用户定制的访问控制策略，负责计算对数据拥有者而言，提起下载请求用户的社交关系度，判断是否与既定的访问控制策略相冲突。**数据管理中心**响应系统用户上传和下载数据的请求，在此过程中，与用户管理中心和访问控制策略管理中心密切合作。例如，当下载数据的请求到达时，首先验证发起请求的用户是否为系统的有效用户，其次判断此次请求是否符合既定的访问控制策略，当且仅当两次验证全部通过时，

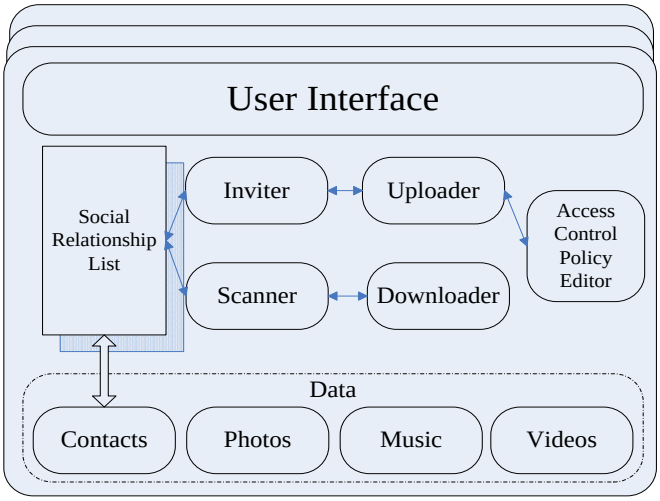

Figure 3 (a) 手机客户端

图 3(a) clients

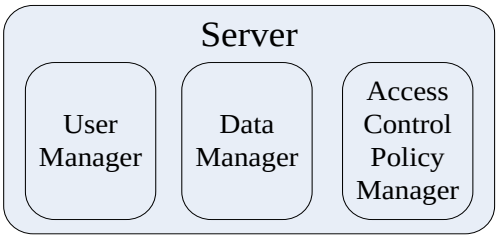

Figure 3 (b) 服务器端

图 3(b) sever

Figure 3 System Architecture

图3 系统架构

2 其中，1 代表该通路途经的结点数；0.5 则是该通路上从 David 出发的社交关系有向边上的社交关系度，即对 Harry 而言 David 的社交关系度(Coworkers，默认社交关系度为 0.5)，

3 之所以引入用户名，而不直接将手机号作为系统用户的唯一标识，是因为用户与手机号之间并不是 1-1 的映射关系。例如，大多数商业人士都拥有多部手机，借以区分私人联系、商务往来等不同的用途。

下载数据的请求才能成功。

3.2 手机客户端

手机客户端相对比较复杂。如图 4 所示，它由社交关系列表(Social Relationship List)、邀请中心(Inviter)、短信扫描服务(SMS Scanner)、上传服务(Uploader)、下载服务(Downloader)和访问控制策略编辑器(Access Control Policy Editor)六个部分组成。

社交关系列表在通讯录的基础上生成。用户为通讯录中的每位联系人打上诸如家人(Family)、密友(CloseFriends)、普通朋友(Friends)、同学(Classmates)、同事(Coworkers)、商务伙伴(Business Partners)等标签。系统根据标签对联系人进行分类，生成社交关系列表，并上传至服务器。值得注意的是，在生活中，存在一些联系人既是朋友也是同学，甚至还是同事。故本系统支持用户为联系人标记多个标签，其社交关系度的计算以最小值为准。考虑到灵活性和可扩展性，本系统还支持用户新增或调整默认的社交关系度设置。

使用**访问控制策略编辑器**，用户可以为亲疏有别的联系人分配不同的访问权限，以便更好的保护数据安全与个人隐私。**邀请中心**负责向相关联系人发送分享数据的邀请。**扫描服务**主动检索到达的邀请，自动跳转到相应的链接，获得用户允许后触发下载服务。

4 平台的实现与实例研究

4.1 平台的实现

我们在 Apache 服务器和 Android G2 手机上搭建了支持基于社交关系度的数据共享与访问控制的原型系统。从 2.2 小节中社交关系度的计算方法可知，这属于经典的“最短路径”问题，可以采用 Dijkstra 算法。考虑到算法复杂度和系统性能方面的约束，我们利用访问控制策略<A, O: O1|O2|…|On, D, R>中的社交关系度 D 的值来减少 Dijkstra 算法的使用频率。当 D 小于 1 时，查找用户 A 的直接联系人以获取 B 的社交关系度；当 D 在 1～2 之间时，判断用户 A 和 B 的直接联系人是否存在交集，并根据交集来计算 B 的社交关系度(取最小值)；当 D 为 6 时，我们推断用户 A 希望尽可能多的人来访问数据，可直接设置 B 的社交关系度为 6；当且仅当 D 的值在 2～6 之间时，才调用 Dijkstra 算法。

4.2 实例研究——通讯录共享

本文以通讯录作为数据共享的一个实例。现代人快速的生活节奏以及频繁的工作变动有可能导致联系方式的不断更换，这使得用户需要经常性地手动更新自己的手机通讯录，造成了极大的不便。

以本文提出的数据共享方式为基础，我们实现了一种集中式的通讯录管理机制。在这种新的共享机制下，系统用户只需在自己的手机通讯录中修改个人信息，并与朋友共享即可。此后的所有数据更新都会以透明的方式传播到所有包含该信息的用户的手机上。另外，该方式还使用户之间可以共享完整的通讯录信息，以扩大社交范围。

随着现代人联系方式的增加，手机通讯录的结构日益复杂，以 Android G2 为例，包含了五个类别十数个条目的信息：联系电话(可扩展为固定电话、办公电话、移动电话等)、电子邮箱(可扩展为办公邮箱、常用邮箱、备用邮箱、私人邮箱等)、通讯地址(可扩展为办公地址、家庭地址等)、即时聊天(可扩展为 MSN、QQ、Gtalk 等)和社交网络(可扩展为 Facebook、Twitter、人人网等)。但是除了极少数非常熟悉的朋友，通讯录中大部分联系人的信息，电话以外几乎都是空白。

值得注意的是，在上文所提到的十数种联系方式当中，对于不同社交关系度的联系人，用户愿意分享的信息量是不同的。例如，对于商业伙伴，用户乐于分享办公电话等与工作相关的联系方式，而不愿提供移动电话、家庭地址等隐私信息；但是对于朋友或者家人，用户则希望提供更多甚至全部的联系信息。利用本文实现的数据共享平台，用户在完善并分享个人信息的同时，可以设置合适的访问控制策略，从而细粒度的控制不同社交关系度的联系人获得的信息量。例如，用户可以选择与所有人共享办公电话（不限社交关系度），只与直接联系人共享手机号码（社交关系度<1），而家庭地址则只共享给关系比较亲密的朋友和家人（例如，社交关系度<=0.3）。

我们在 Android G2 手机上实现了用户通讯录工具，并通过网络(WiFi, GPRS, 3G 等)把数据保存到服务器上，实现相关数据的透明共享。以图 1 所示的社交关系为例，用户 Bella 在本平台上分享了她的个人信息，并设置访问控制策略使得密友级别的联系人和普通朋友级别的联系人获得不同的信息量。作为密友，用户 Edward 的通讯录如图 4 所示；而作为普通朋友，用户 Angela 的通讯录如图 5 所示。

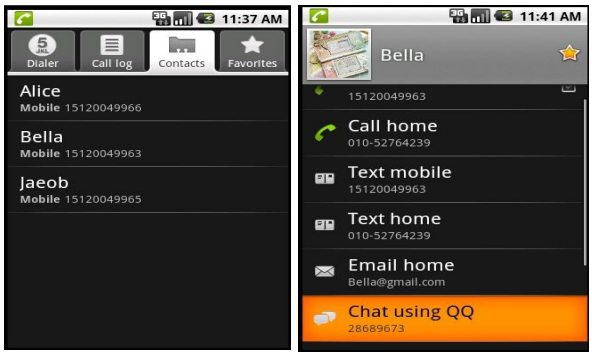

Fig. 4 Edward’s Contacts
图 4 Edward 的通讯录

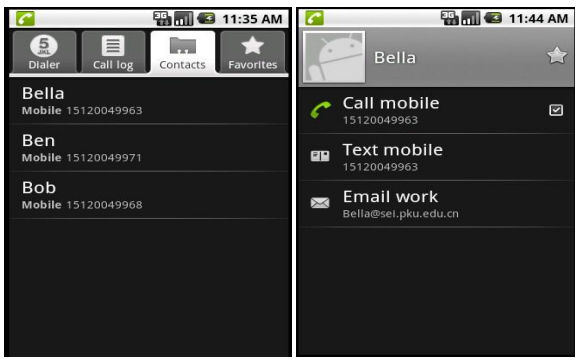

Fig.5 Angela’s Contacts
图 5 Angela 的通讯录

在不同的信号强度下，用户 Edward 通过本平台获得 Bella 的个人信息所需时间如表 1 所示。

Table 1 Time Taken under Different Signal Intensity(ms)
表1 不同信号强度下的耗时(毫秒)

Signal Intensity	No.1	No.2	No.3	No.4	No.5	Average[4]
Good	1828	1767	1805	1797	1787	1796.3
Middle	2503	2065	2290	1978	2107	2154
Bad	3062	3766	4042	3577	3689	3677.3

5 相关工作

社交网络为人们提供了方便的数据共享平台，人们可以通过社交网络来分享个人动态、照片、日志、音乐和视频等数据。随着社交网站的普及和大型社交网站的崛起，数据安全和个人隐私的重要性日益凸显。

为了使用户既能享受到便捷的数据共享服务，又可以保障数据的安全，Stanford 大学的 Monica Lam 教授及其团队开发了名为 PrPl[2-4]的项目，提出了应用与数据分离的概念。该项目主张在应用程序和数据之间插入一个通用的数据存取平台，这个平台向上为应用程序提供统一的数据存取接口，向下管理存储在云端或其他用户指定存储设备上的用户数据。应用程序访问数据的请求均需要通过该平台的有效性验证，从而保障数据的安全。通用的数据存取平台使得用户跨越不同的虚拟社交网络分享数据成为可能。

相似的，MIT 的 Ramesh Chandra 等人也开发了一个名为 BSTORE[5]的项目，实现了应用与数据的分离。在核心理念上，BSTORE 与 PrPl 有着非常相似之处，但是在具体实现的细节上有所不同。例如，为了方便上层的应用程序存取数据，PrPl 提供了一套名为 SociaLite 的数据库查询语言，而 BSTORE 则提供了一套文件系统的 API。此外，PrPl 面向社交网络的用户，旨在使用户可以跨越不同的虚拟社交网络分享数据；而 BSTORE 则是面向网络应用程序的开发者，为他们减轻数据管理的重担。

由此可见，采用通用的数据共享机制来隔离应用程序和数据，可以为用户和应用程序的开发者同时带来便利。本文在上述两项工作的启发下，将通用的数据共享与访问控制机制在移动平台上予以实现，以期为手机用户(以及上层的应用程序)提供一种安全、透明的通用数据共享方式。

为了给用户提供更及时、更快捷的数据共享服务，移动环境下的数据共享方式也成为研究的热点。MIT 的 Bryan Ford 等人开发的项目 UIA[6-7]，将用户的台式机、笔记本、手机、数码相机等个人设备组织起来，用户为每个设备赋予一个独一无二的名字，这些设备之间即使跨越公网也可以根据名字来彼此寻址，从而使用户能够方便的进行跨设备的数据共享。经过改进之后的 UIA 还可以支持不同用户不同设备之间的数据共享。值得注意的是，UIA 要求参与通讯的两个节点必须同时在线，而本文提出的数据共享机制，对于数据接收者的状态并无限制。

在共享数据的同时，为了保障数据安全，通常采用访问控制机制。Barbara Carminati[8]等人与 Bader Ali[9]等人都提出了基于信任度的访问控制机制。这套机制根据系统中所有用户对某一用户的信任程度，计算该用户的信誉值，并据此建立访问控制策略。Barbara Carminati 等人在类似分布式系统的环境下实现了该机制。而 Bader Ali 等人把这套机制引入了 flickr.com。

6 总结与未来的工作

本文提出了一种基于社交关系度的数据共享与访问控制机制，使得用户能够更清晰的阐述自身的共享需求、更细粒度地划分数据的等级，从而灵活地控制数据(或部分数据)的流向、更好地保障数据的安全。本文在 Apache 服务器和 Android G2 手机上实现了这套机制的原型系统。并以通讯录的共享为例进行了验证。

在后续的工作中，我们将进一步完善这套机制，以支持照片、音乐等多种类型数据的共享，把移动社交网络打造成安全便捷的数据共享平台。我们还将向笔记本等移动设备及台式机扩展，以期开发一套跨设备跨平台的通用数据共享机制。此外，我们还将进一步探索如何在分布式环境下搭建系统并使之顺畅运行。

7 参考文献

[1] *Six degrees of separation*[OL].Available: http://en.wikipedia.org/wiki/Six_degrees_of_separation

[2] S.-W. Seong, *et al*. (2009, *The Architecture and Implementation of a Decentralized Social Networking Platform*[C].

[3] S.-W. Seong, *et al*. (2009, *Preserving Privacy with PrPl: a Decentralized Social Networking Infrastructure*[C].

[4] S.-W. Seong, *et al*., "PrPl: a Decentralized Social Networking Infrastructure[C]," in *In Proceedings of the 1st International Workshop on Mobile Cloud Computing & Services: Social Networks and Beyond*, San Francisco, 2010.

[5] R. Chandra, *et al*., "Separating Web Applications from User Data Storage with BSTORE[C]," presented at the 1st USENIX Conference on Web Application Development (WebApps '10), Boston, Massachusetts, 2010.

[6] B. Ford, "UIA: A Global Connectivity Architecture for Mobile Personal Devices[D]," Doctor of Philosophy, Electrical Engineering and Computer Science, Massachusetts Institute of Technology, 2008.

[7] B. Ford, *et al*., "Persistent personal names for globally connected mobile devices[C]," presented at the Proceedings of the 7th symposium on Operating systems design and implementation, Seattle, Washington, 2006.

[8] B. Carminati, *et al*., "Enforcing access control in Web-based social networks[C]," *ACM Trans. Inf. Syst. Secur.*, vol. 13, pp. 1-38, 2009.

[9] B. Ali, *et al*., "A Trust Based Approach for Protecting User Data in Social Networks[C]," presented at the In 2007 Conference of the Center for Advanced Studies on Collaborative research(CASCON'07), 2007.

4 分别去掉一个最优的数据和一个最差的数据，取剩余三次数据的平均值。