基于 P2P 的无需第三方验证的本地信誉存储管理

孙 华[1,2],虞慧群[1,3],杨年华[1]
(1.华东理工大学计算机科学与工程系,上海 200237; 2.新疆大学信息科学与工程学院,新疆乌鲁木齐 830046;
3.上海市计算机软件评测重点实验室,上海 201112)

摘 要: 目前将信誉信息存储在本地进行管理的方法需要通过第三方验证信誉的完整性,这在动态的 P2P 环境中很难实现,而且信誉信息易丢失或被篡改.提出了一种新的本地存储信誉信息的管理方法,将信誉信息以证书的形式存储在本地,并由所有者自行维护.通过设计相应的协议,实现了旧信誉证书的完整性验证以及新信誉证书的生成.该方法避免了大量的通信开销,提高了查询效率,并具有一定的安全性.

关键词: Peer-to-Peer; 信誉证书; 数字签名; 存储; 完整性
中图分类号: TP393 **文献标识码:** A **文章编号:** 0372-2112 (2011) 3A-104-06

Local Storage of Reputation Without the Third Parties to Validate Integrity in Peer-to-Peer Environments

SUN Hua[1,2], YU Hui-qun[1,3], YANG Nian-hua[1]
(1. *Department of Computer Science and Engineering, East China University of Science and Technology, Shanghai* 200237, *China*;
2. *School of Information Science and Engineering, Xinjiang University, Urumqi, Xinjiang* 830046, *China*;
3. *Shanghai Key Laboratory of Computer Software Evaluating and Testing, Shanghai* 201112, *China*)

Abstract: There are some methods to store reputation locally to improve efficiency, but it needs the third parties to avoid the information dropped or tampered by the peers themselves. It's difficult to take it into reality in distributed Peer-to-Peer environments. We propose a novel protocol to store the reputation locally, which does not need the third parties. The reputation is stored in the certificates of the peers and maintained by themselves. The corresponding protocol realizes the integrity validation and creates the reputation certificates at the same process. This method can reduce communication overhead and improve the query efficiency. It's also resistant to several of attacks.

Key words: Peer-to-Peer; reputation certificates; digital signature; storage; integrity

1 引言

在分布式 Peer-to-Peer 网络中,信任和信誉已经出现并作为重要的决策支持工具去选择在线的服务以及评估获取服务的风险[1].信任与信誉模型被提出来以解决 P2P 环境下信任和信誉的查询和评估[2~5].待交易的节点为了全面地评估交易对方节点的信誉,需要其它节点的推荐[6],这在分布式 P2P 环境中[7]往往导致大量的传播消息,如 P2P 下的应用系统 Gnutella[8] 使用泛洪(flooding)的方式以及其它的应用于 P2P 环境下的信誉查询系统[9]等.

信任与信誉系统的最基本的思想就是让交易双方互评[10],评价信息可以帮助其它节点决定是否与被评价节点进行交易.例如,EigenRep[11]采用分布式安全迭代计算方法以阻止恶意节点.信任值的存储位置直接影响着信誉的查询与计算效率.如果存储在被评价者本地[12],其它节点可以直接查询,不仅提高检索效率,而且可避免恶意节点的错误反馈信息[13],但是信誉信息的完整性将面临很大的挑战.目前将信誉信息存储在本地的管理方法,需要通过第三方(TTP, The Third Parties)进行完整性验证[14,15].但是在分布式 P2P 网络中,大量的节点频繁地加入和离开网络,在验证完整性时,很难保证第三方证人是在场的,或者节点已经在查询的边界之外[16].

收稿日期:2010-10-28;修回日期:2011-03-09
基金项目:国家自然科学基金(No.60473055,No.60773094)及上海市曙光计划(No.07SG32)

针对以上问题,本文提出了一种新的本地存储信誉的管理方法,将节点的信誉以证书的形式保存在本地,并由节点自行维护和更新.每次交易前,节点将自己的信誉证书交给对方验证完整性;交易结束后,由交易双方互相评价并签名,生成相应的信誉证书.其它节点仅通过该节点就可以得到其它节点对该节点的全面的评价信息,而且对其证书的完整性验证也很容易实现,不需要第三方的参与.本文通过设计相应的协议,实现了信誉证书的生成以及完整性验证,并从安全性和性能等方面进行了分析和比较.

2　相关研究工作

Ooi 等[14]在 2003 年提出了一个分布式 P2P 环境下的信誉管理系统,采用信誉证书 RCert 记录其它与之交易过的节点对其行为所做的评价.引入了公钥加密体制(PKI, Public Key Infrastructure),利用 PKI 和建立在上面的一些机制,保障信息的完整性.当验证信誉数据的真实性时,要求上一次的交易对方作为证人在场,从而限制了协议的可用性.Lintao Liu 等[15]在 RCert 的基础上提出一个轻量级可扩展的信誉管理系统 R-chain.该系统采用监管的形式,随机选择几个证人进行监管并对评价进行签名.但是在处理如何选择证人以及证人共谋的情况时,该文处理的非常繁琐.另外 R-chain 没有设计表头,不能避免整个链表被删除的情况.TrustMe[17]是由 Aameek Singh 等提出来的 P2P 分布式的环境下信任关系的匿名管理系统.该系统也采用 PKI 机制,并利用多个密钥对实现安全管理.但是该系统有一些完全被信任的引导服务器,在完全分布式的环境中难以实现.hiREP[18]是一个分级的信誉管理系统,采用分级的体系结构,解决信任值的存储和发布,同时保证将真实的交易结果报告给信誉代理.文献[19]提出一个可信的信誉管理服务(TRMS),采用分布式的方法聚合和发布节点的信誉,能够保证系统的安全性、高效和扩展性.Hao L 等[20]提出了一个高效健壮的自我存储的 P2P 信誉系统,它提供了高效的检索节点评级的一种方法.每个节点都将其它节点对于它的正面的评级存放在本地.Prashant Dewan 等[21]研究了 P2P 网络中的加强信誉数据安全的方法,提出了一个信誉模型以及信誉交换协议.文献[22]中主要描述了超级节点网络下的信任管理结构.下面详细介绍无需第三方的本地信誉存储管理方法.

3　信誉证书生成及验证

3.1　信誉证书格式

信誉证书格式由两部分组成:信誉证书头(RCH, Reputation Certificates Head)和信誉证书(RC, Reputation Certificates).如表 1 所示,信誉证书头包含节点的基本信息.节点的 *ID* 可能会有一个或多个.每个节点拥有唯一的一对公钥 PK 和私钥 SK 分别用于签名和验证,这一对公私钥也用于用户的身份识别.TID_0 是第一个信誉证书交易号(TID_1)的前驱号码,用来防止恶意节点将自己的信誉证书全部删除(详细见 3.3 节).OtherInfo 可以根据需要定义.

表 1　信誉证书头的内容

内容	符号
节点标识	*ID*
节点公钥	PK
第一个交易号的前驱号码	TID_0
其它信息	OtherInfo

信誉证书记录着节点的评价信息,每个信誉证书代表一次评价,包含的内容如表 2 所示.证书中的评价内容由对方节点评出并进行签名.每笔交易都会在消费节点和服务节点中分别产生一个信誉证书(生成过程见 3.2 节),下面从一笔交易中节点所承担的角色分别是服务节点和消费节点的角度,详细说明其中的内容:

表 2　信誉证书的内容

内容	符号
交易号	*TID*
时间	TS
角色标志位	RF
节点对交易号、时间和角色标志位的签名	Signature
交易对方的公钥	PK
交易对方对本次交易的评价	Rating
交易对方的签名	Signature
其它信息	OtherInfo

(1)对于消费节点 *C*

当 *C* 请求服务时,产生一个空的信誉证书 $RC_{C-New,i}$,表示准备开始第 i 笔交易.交易号 $TID_{C,i}$由系统生成,是节点交易信息的唯一标识,由前后两部分组成:$TID_{C,i}$的前缀 $Prefix(TID_{C,i})$和 $TID_{C,i}$的后缀 $Postfix(TID_{C,i})$,前缀由公钥的单向函数产生:$Prefix(TID_{C,i}) = Hash(PK_C)$;后缀由公钥做为种子,产生一个连续的编码.为了保证所有交易号的连续性,编码器中需要设置一个记忆单元,记录上一次产生的编码,以保证每次生成不重复的连续编码.因为各个节点的公钥是不同的,所以同一笔交易中的交易双方证书中的交易号是不同的.时间 $TS_{C,i}$是系统时间.角色标志位 $RF_{C,i}$为消费.*C* 用自己的私钥签名处理以上信息后发送给服务节点 *S*.当交易结束后,由 *S* 对 *C* 评价并用自己的私钥签名.证书中的其它信息 $OtherInfo_{C,i}$可以根据需要定义.

(2)对于服务节点 *S*

如果 *S* 接受消费节点 *C* 的请求,假设将要进行第 k 笔交易.*S* 会生成一个空的信誉证书 $RC_{S\text{-}New,k}$,交易号 $TID_{S,k}$与时间 $TS_{S,k}$的生成同 *C* 的生成过程.角色标志

位 $RF_{S,k}$为服务.经过 S 签名的新信誉证书发送给 C.

在同一笔交易中,C 和 S 分别产生的两个证书的时间不要求完全一致,它们只要满足式(1)即可,其中 τ 为双方节点可以接受的时间延迟.否则一方或者双方重新生成新的证书,原来的新证书做为空证书仍然保留.

$$| TS_{S,k} - TS_{C,i} | \leqslant \tau \quad (1)$$

交易结束后,由 C 对 S 评价并用自己的私钥签名.

3.2 信誉证书生成及验证协议

3.2.1 协议中的符号说明

假设网络中的各节点即将进行交易,按一笔交易中所承担的角色进行划分,提供服务的节点称为服务节点 S,享用服务的节点称为消费节点 C,所有能够提供本次交易中指定服务的节点称为资源节点 R.这些节点的角色划分是动态变化的.假设 C 即将进行第 $n+1$ 笔交易,S 即将进行第 $m+1$ 笔交易.交易中的一些证书及符号说明如下(以消费节点为例,其中 ‖ 为连接符,下同):

证书头:$RCH_C = ID_C \parallel PK_C \parallel TID_{C,0} \parallel OtherInfo_C$;

旧信誉证书(前 n 笔交易产生的证书):$RC_{C\text{-}Old} = RC_{C,1} \parallel RC_{C,2} \parallel \cdots \parallel RC_{C,i} \parallel \cdots \parallel RC_{C,n}$;

第 $n+1$ 笔交易前新产生的证书:$RC_{C\text{-}New,n+1} = TID_{C,n+1} \parallel TS_{C,n+1} \parallel RF_{C,n+1} \parallel Sign_{SKc}(TID_{C,n+1} \parallel TS_{C,n+1} \parallel RF_{C,n+1})$;

第 $n+1$ 笔交易产生的证书:$RC_{C,n+1} = RC_{C\text{-}New,n+1} \parallel PK_S \parallel Rating_{C,n+1} \parallel Sign_{SKs}(RC_{C\text{-}New,n+1} \parallel PK_S \parallel Rating_{C,n+1})$;

验证 S 的旧信誉证书:$Verify\_Sequence(RC_{S\text{-}Old})$.

3.2.2 协议设计

协议过程如图 1 所示,包括新信誉证书的生成以及旧信誉证书的验证.

交易结束后,将刚产生的信誉证书加入到旧信誉证书中,C 与 S 分别更新旧信誉证书的内容:

$$RC_{C\text{-}Old} = RC_{C\text{-}Old} \parallel RC_{C,n+1}$$

$$RC_{S\text{-}Old} = RC_{S\text{-}Old} \parallel RC_{S,m+1}$$

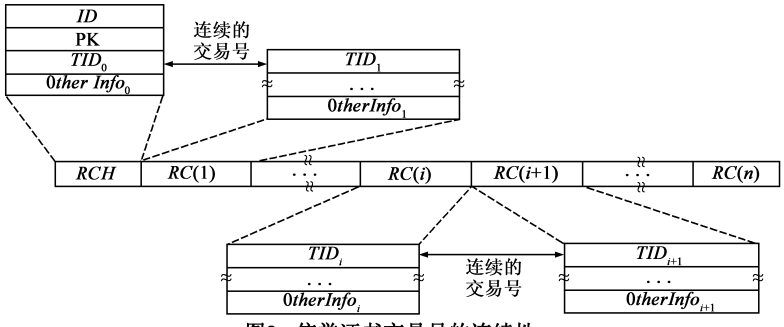

图2 信誉证书交易号的连续性

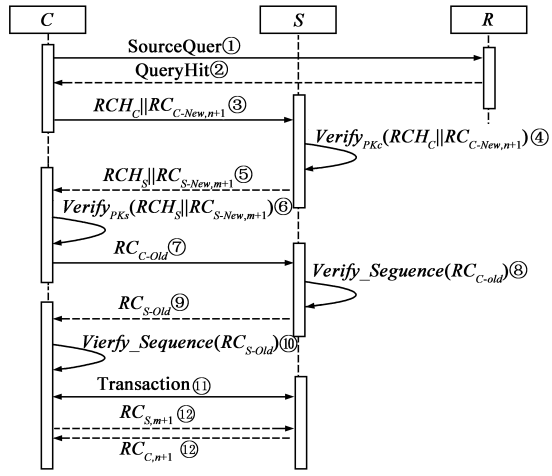

图1 完整性验证过程以及证书生成过程

3.3 过程分析

下面针对 3.2 节协议中的几个问题进行讨论:

(1)验证过程的对等性

交易双方均需要经过两次验证,实现对方的信誉证书的完整性验证,双方地位的对等性非常符合 P2P 的特点.如果任何一方在验证的过程中发现对方是不可靠的,都可以通知对方终止协议.

(2)交易号的验证

双方在验证如图 2 所示,交易号可能存在以下问题:

(a)错误的交易号

根据交易号与公钥之间的关系,其它节点验证信誉证书的真伪.如果验证失败,那么该证书不属于该节点.

(b)交易号的不连续性

交易号的连续性如图 2 所示.证书的内容经过签名后,如果想删除不良记录,证书所有者只能将一个或若干个信誉证书整体删除.通过验证信誉证书中的 TID_0 与 TID_1 之间,或者 TID_i 和 TID_{i+1}($1 \leqslant i < n$)之间的连续性,从而验证是否存在删除现象.

(3)签名的验证

节点不能删除证书中的若干个或者全部证书.为了防止信誉证书中的内容被修改,如复制其它节点信誉证书,或将一个不良的记录删除形成一个空证书,要求对方节点在信誉证书上签名,签名的范围是信誉证书中除其它信息以外的所有信息项.通过证书中交易对方的公钥,可以方便地检测出是否被篡改过.

(4)时间和空证书

交易双方在交易前就已经产生了新

的空信誉证书,并产生了相应的时间.如果协议被提前终止,消费节点需要再次选择下一个服务节点,根据式(1),时间很有可能已经过了对方的容忍延迟,那么这个信誉证书成为一个空证书,消费节点将再次生成一个新的空信誉证书请求服务;服务节点也是一样.这些空证书也需要保留,以保证连续的交易号,它们并不会影响节点的信誉.

(5)系统维护

信誉证书的系统维护工作也非常重要,在本文中涉及到两个问题:(1)如何处理过期(间隔时间比较久)的信誉证书;(2)如何设置证书的存储结构,方便系统的维护工作.因为有时其它节点并不需要查询所有的信誉信息,而且有些经常交易的节点,信誉信息更新非常快,造成本地庞大的信誉证书难以维护.本文采用循环队列(cyclic queue)的存储结构,如图 3 所示.循环队伍中包含各信誉证书和信誉证书头.循环队列的大小由节点加入到网络时根据需要进行定义.按照队列"先进先出"的特点,在循环队列中,较老的信誉信息被新的信息覆盖.当队列未满时,其它节点可以按照前面的方法验证节点信誉信息的完整性.如图 4(a)中所示,但是,当队列已经满时,再插入进来的信誉信息将依次覆盖最早产生的信誉证书,如图 4(b)所示.因此,队伍中保存了节点最近所从事的交易信息,验证完整性只需要验证整个队列中所存储的信誉证书的连续性.

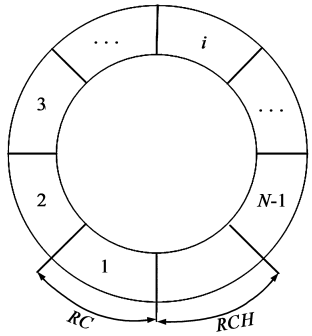

图3　循环队列存储结构

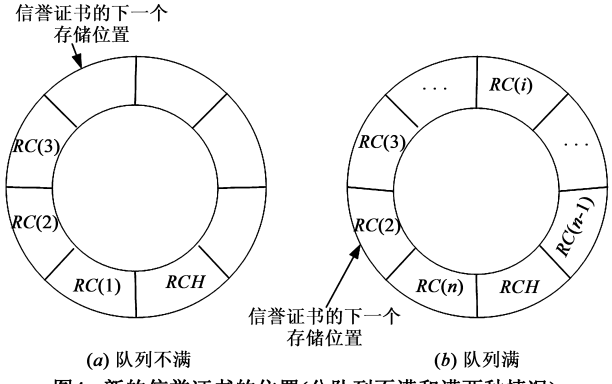

图4　新的信誉证书的位置(分队列不满和满两种情况)

4　安全性分析

P2P 网络开放、匿名和动态特性使其面临很多安全威胁,本文所设计的协议从安全方面做了详细的考虑,具体分析如下:

(1)　防止搭车行为(free riding)

通常情况下,信誉管理中的服务节点的信誉可能会受到更多的关注.而在本协议中,服务节点和消费节点受到同样的关注.通过设置角色标志位 RF,可以统计节点享用服务的次数和提供服务的次数,见式(2),其中 λ 是资源节点能够提供服务的阀值,由服务节点确定具体的值.

$$\frac{\text{享用服务次数}}{\text{提供服务次数}} = \frac{\sum_{RF=C}}{\sum_{RF=S}} \geqslant \lambda \qquad (2)$$

(2)保密通信

如果交易的双方需要保密通信,C 可以在第三步同时发送一条保密请求的消息 RequestMessage(Encryption),S 收到后,在第五步,用 PK_C 加密一个对称加密的密钥,假设为 K_{CS},发送给 C,即 ReplyMessage (E_{PKc} (K_{CS})). C 使用自己的 SK_C 解密,即 $D_{SKc}(E_{PKc}(K_{CS}) = K_{CS}$.在后面的通信过程中,双方可以用 K_{CS}加密信息后再发送给对方.该协议中使用的是对称密钥,如果发送的信息量很少,也可以使用对方的公钥进行加密处理,收方用私钥解密.

如果将协议中的第三步与第七步交换,第五步与第九步交换,即先发送旧的信誉证书,再发送新产生的空信誉证书,因为协议的执行并不能总是完全的,所以相对比本文所采用的方式,需要更多的通信量以及计算量.

5　性能分析

5.1　实验与性能对比

本文介绍的将信誉信息存储在本地,并由交易对方验证信誉信息完整性的方式,既提高了查询的效率,又不用第三方介入.这种方式跟现有的信誉信息管理方法相比,极大地减少了查询的时间,而且信誉证书的维护也很简单.图 5 显示了没有信任关系的情况、R-chain 和文献[20](以下各图中标为 Hao system)的对比结果.

通过对以下几种信誉管理进行分析:集中式信誉管理,如 eBay,分布式信誉管理,如 RCertP[14] 和 R-

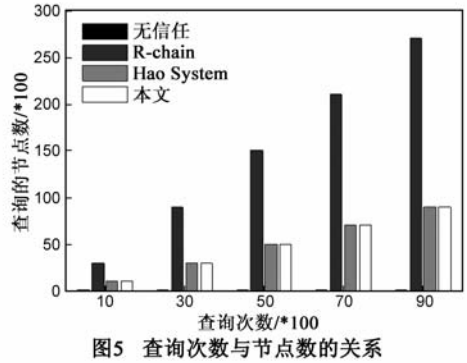

图5　查询次数与节点数的关系

chain[15],以及 P2P 下普通分布式信誉查询管理,如 EigenRep[11]和 Hao[20].其中 RCertP、R-chain、Hao system 和本文都是将信誉信息存储在本地.本文从交易的失败率,分析了这几种信誉管理系统的性能对比.假设节点数目为 1000,其中诚实节点的百分比为 80%到 20%,相应的恶意节点的百分比为 20%到 80%,节点的交易率为 75%,恶意节点从事恶意行为的百分比为 80%,诚实节点从事诚实行为的百分比为 90%,参与的交易数目为 1000 次.恶意节点百分比与交易失败率如图 6 所示.为了防止恶意节点,R-chain 和本文都做了大量的安全分析,所以,从图 6 中可以看出,失败的交易百分比变化比较小.如果没有信任管理,那么当恶意节点的百分比比较高时,失败的交易率非常高.

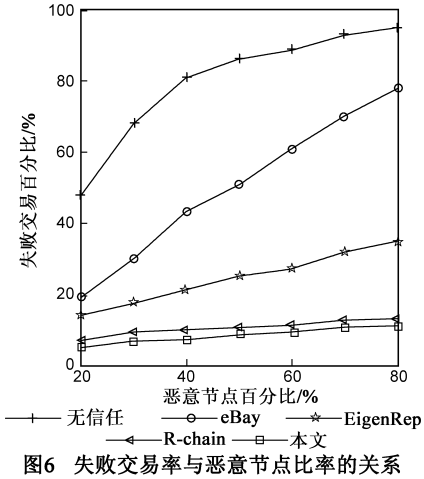

图6 失败交易率与恶意节点比率的关系

本文方案无需第三方节点的验证以及其它的节点的加入.在 Hao 提出的系统中,也只需要交易的双方节点,而其它信誉管理系统中,因为每笔交易都需要第三方节点的加入,当网络规模增加时,每笔交易中所涉及的节点数都会急剧增加.如图 7 所示,当网络规模增加时,所涉及的节点数也会增加,网络开销非常大.

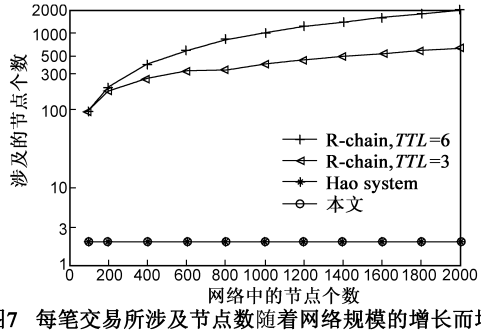

图7 每笔交易所涉及节点数随着网络规模的增长而增加

如果用响应时间代表查询节点从查询到收到足够判断一个节点信誉的信任信息的时间,那么在 TrustMe[17]系统中,每个节点的信誉值被保存在多个节点中,查询的响应时间比本文方法所消耗的响应时间多,如图 8 所示.虽然 TrustMe 中可以采用多个 THA 的方式减少查询的响应时间,但是从图中可以看出累加的响应时间比本文的方法多 62%.

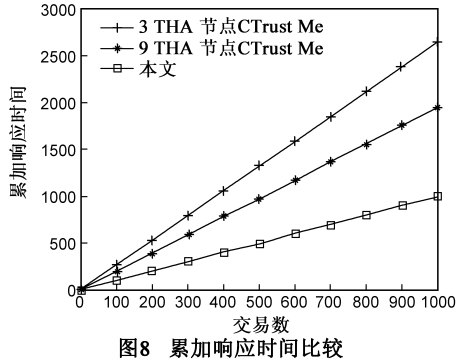

图8 累加响应时间比较

5.2 扩展性

本文前言中提到一些计算或者评估信誉的模型,这些模型在处理信誉值时需要先查询节点的信誉,再计算信誉值.本文所设计的方法不仅提高了信誉查询的效率,而且没有包含任何前提假设,没有做一些特殊处理,所以可以与任何一个模型有效的结合,从而快速全面地得到一个节点的信誉值.

6 结论

本文提出将信誉信息保存在本地并由交易对方验证完整性的方法,可以避免通信开销,能保证信誉信息的完整性,而且不需要证人或者权威机构的参与.同时,协议具有较高的安全性以及较好的扩展性.采用信誉证书证明信息完整性的方法,虽然不用通过权威机构颁发,由所有者节点自己生成并由对方节点签名,但是如果交易节点的私钥丢失或被盗,那么证书的真实性将存在问题,需要作废处理.另外节点刚加入到网络中,并不能很好地预测出循环队列的大小,以确定存储的开销.因此如何动态地处理循环队列的大小,一方面不影响完整性验证,另外又可以保证信誉信息的充分性,需要进行深入的研究.

参考文献

[1] A Josang, T Bhuiyan, Y Xu, et al. Combining trust and reputation management for web-based services[A]. Trust, Privacy and Security in Digital Business, Lecture Notes in Computer Science [C]. Berlin Heidelberg: Springer-Verlag, 2008, 5185. 90 – 99.

[2] A Josang, R Ismail. The beta reputation system[A]. 15th Bled Electronic Commerce Conference e-Reality: Constructing the e-Economy[C]. Bled, Slovenia, 2002. 324 – 337.

[3] L Mui, M Mohtashemi, A Halberstadt. A computational model of trust and reputation[A]. Proceedings of the 35th Hawaii international conference on system science[C]. Los Alamitos: Ieee Computer Soc, 2002. 280 – 287.

[4] T D Huynh, N R. Jennings, N R Shadbolt. An integrated trust

and reputation model for open multi-agent systems[J]. Auton Agent Multi-Agent Sys,2006,13(2):119 – 154.

[5] 赵翔,黄厚宽,董兴业,等.开放多 Agent 系统的一个信任信誉系统模型[J].计算机研究与发展,2009,46(9):1480 – 1487.
Zhao Xiang,Huang Houkuan,Dong Xingye,He Lijian. A trust and reputation system model for open Multi2Agent system[J]. Journal of Computer Research and Development,2009,46(9):1480 – 1487.(in Chinese)

[6] 李景涛,荆一楠,肖晓春,等.基于相似度加权推荐的 P2P 环境下的信任模型[J].软件学报,2007,18(1):157 – 167.
Li Jing-tao,Jing Yi-nan,Xiao Xiao-chun,et al. A trust model based on similarity-weighted recommendation for P2P environments[J]. Journal of Software,2007,18(1):157 – 167.(in Chinese)

[7] R Spanek,M Rimnac,Ieee. The reputation system for distributed data source environment[A]. 1st International Conference on the Applications of Digital Information and Web Technologies [C]. Ostrava,2008.495 – 500.

[8] T Klingberg,R Manfredi. Gnutella Protocol Development June 2002[DB/OL]. http://rfc-gnutella. sourceforge. net/src/rfc-0_6-draft.html.

[9] Cornelli F,Damiani E,di Vimercati SDC. Choosing reputable servents in a p2p network[A]. Proceedings. of the 11th International World Wide Web Conference[C]. New York:ACM Press.2002.376 – 386.

[10] A Josang,R Ismail,C Boyd. A survey of trust and reputation systems for online service provision[J]. Decision Support System,2007,43(2):618 – 644.

[11] S D Kamvar,M T Schlosser,H Garcia-Molina. EigenRep:reputation management in P2P networks[A]. Proceedings of the 12th International World Wide Web Conference[C]. New York:ACM Press.2003.123 – 134

[12] P Dewan. Peer-to-Peer Reputations[A]. Proceedings of the 18th International Parallel and Distributed Processing Symposium[C]. New Mexico:USA,2004.128.

[13] Jin Y,Gu ZM,Gu JG,et al. A new reputation-based trust management mechanism against false feedbacks in peer-to-peer systems[A]. Proceedings of the 8th international conference on Web Information Systems Engineering (WISE 2007) [C]. Berlin Heidelberg:Springer-Verlag,2007.62 – 73.

[14] B C Ooi,C Y Liau,K-L Tan. Managing trust in peer-to-peer systems using reputation-based techniques[A]. Proceedings of the 4th International Conference on Advances in Web-Age Information Management,Lecture Notes in Computer Science [C]. Berlin Heidelberg:Springer-Verlag,2003,2762.2 – 12.

[15] L Liu,S Zhang,K D Ryu,et al. R-chain:A self-maintained reputation management system in P2P networks[A]. 17th International Conference on Parallel and Distributed Computing Systems[C]. San Francisco,CA,2004.131 – 136.

[16] 金瑜,古志民,顾进广,等.一种对等网中基于相互信任的两层信任模型[J].软件学报,2009,20(7):1909 – 1920.
Jin Y,Gu ZM,Gu JG,Zhao HW,et al. Two-level trust model based on mutual trust in peer-to-peer networks[J]. Journal of Software,2009,20(7):1909 – 1920.(in Chinese)

[17] A Singh,L Liu. TrustMe:Anonymous management of trust relationships in decentralized P2P systems[A]. The Third International Conference on Peer-to-Peer Computing (P2P'03) [C]. Linkoping:Sweden,2003.142 – 149.

[18] X M Liu,L Xiao. hiREP:Hierarchical reputation management for peer-to-peer systems[A]. Proceedings of the International Conference on Parallel Processing[C]. Los Alamitos:Ieee Computer Soc,2006.289 – 296.

[19] L Deng,Y He,Z Xu. Trusted reputation management service for peer-to-peer collaboration in the move to meaningful internet systems[A]:OTM 2008[C]. Berlin Heidelberg:Springer Berlin,2008.1069 – 1086.

[20] L M Hao,S N Lu,J H Tang,et al. An efficient and robust self-storage P2P reputation system[J]. International Journal of Distributed Sensor Networks,2009,5(1):81 – 88.

[21] P Dewan,P Dasgupta. Securing reputation data in peer-to-peer networks[A]. International Conference on Parallel and Distributed Computing and Systems (PDCS2004)[C],MIT Cambridge,USA,2004.1 – 6.

[22] T Dimitriou,G Karame,I Christou. SuperTrust-A secure and efficient framework for handling trust in super peer networks [A]. ICDCN 2008[C]. Berlin Heidelberg:Springer-Verlag,2008.350 – 362.

作者简介

孙　华　女,1977 年出生于新疆喀什,分别于 2000 年和 2005 年在新疆大学获得学士和硕士学位,现为华东理工大学博士生,主要研究方向:信誉管理、信息安全.
E-mail:xj_sh@163.com

虞慧群　男,1967 年出生于江苏,教授,博士生导师,中国计算机学会高级会员,主要研究方向为软件工程、信息安全和形式化方法.
E-mail:yhq@ecust.edu.cn