分类号 TP393 学号 06069076

U D C 密级 公 开

工学博士学位论文

信任管理的策略表示与量化模型研究

博士生姓名 王小峰

学科专业 计算机科学与技术

研究方向 网络计算与安全

指导教师 苏金树 教授

Dr. Rajkumar Buyya

国防科学技术大学研究生院

二〇〇九年十月

论文书脊

信任管理的策略表示与量化模型研究

国防科学技术大学研究生院

On the Policy Description and Quantification Model for Trust Management

Candidate：Wang Xiaofeng

Supervisor：Prof. Su Jinshu & Dr. Rajkumar Buyya

A dissertation

Submitted in partial fulfillment of the requirements

for the degree of Doctor of Engineering

in Computer Science and Technology

Graduate School of National University of Defense Technology

Changsha，Hunan，P.R.China

October，2009

目　录

图 目 录

摘 要

人类社会越来越多地依赖于基于互联网的虚拟信息系统。电子商务、电子政务和电子社交等新的互联网应用得到广泛部署；各种新型分布式计算技术如 P2P、Grid 以及云计算等，也使得互联网应用更加简单高效。然而互联网在给人们带来便利的同时，也给人类社会以及人机交互的信任关系带来巨大挑战。面对陌生的用户和无法保证的服务行为，互联网的信任风险正急剧增大。因此，一个可信的计算环境已成为互联网应用拓展的重要基础。

为了对互联网的信任关系进行有效管理，本文根据应用的需求层次将信任管理分为：基于身份的信任、服务属性信任、面向可靠性的信任和面向健壮性的信任。针对前三个层次的信任关系，我们提出紧密相关的四个研究问题：如何设计功能强大的基于身份的信任策略描述语言；如何得到全面且健壮的服务属性信誉模型；如何结合基于策略和信誉的两种信任管理；如何应用信任关系来提高复杂网络应用的可靠性。围绕解决这四个信任问题，本文主要的研究工作和成果是：

(1) 提出一种支持分布式证明和协商的信任策略描述语言和方法

现有的信任协商语言难以支持复杂的访问控制策略和协商策略，对信任分布式证明方法的支持也不够充分。本文提出一种面向信任分布式证明和协商的策略语言 RTP(Role-based Trust Proving)。RTP 通过对 RT 语言进行拓展，可以定义复杂的角色；增加 lsign 语法，能够定义逻辑推导角色并支持信任分布式证明；增加 release 谓词，可以保护信任证书的敏感信息；增加信任协商启发式规则，从而避免信任证书的盲目搜索。文章详细阐述了 RTP 语言的语法构成，定义了 RTP 语言的推理证明规则，给出了语言的语义解释并证明了语言的可靠性和完全性。基于 RTP 语言，进一步提出一个信任分布式证明协商算法 DPN(distributed proving and negotiation)。DPN 通过本地信任协商和远程信任证明，可以高效地完成信任分布式证明任务。文章利用信任图概念分析了算法的正确性和完整性。实验表明，与传统的信任协商方法相比，DPN 算法能够有效地减少信任建立时间和交互次数。

(2) 设计了一个全面且健壮的基于信誉的信任评估模型

基于信誉的信任管理在分布式系统中正变得越来越重要。尽管信誉是信任概率的一个统计估计值，但现有大多数信任模型没有考虑信誉估计偏差；此外大多数研究工作采用相加方法聚合信誉反馈，容易遭受恶意反馈的攻击，且难以增强模型的健壮性。本文提出一个健壮的线性马尔科夫 RLM (Robust Linear Markov)信誉模型，它的特点是通过对信誉评估偏差进行预测，从而能够得到更全面和健壮的信誉评估。RLM 模型将信誉定义成两个参数：信誉估计值和信誉估计方差。为

了聚合信誉反馈，我们提出新颖的卡尔曼反馈聚合方法，它能够支持健壮的信誉评估。为了使模型能够抵御恶意反馈攻击，我们首先采用 EM (Expectation Maximization)算法自主校准模型的动态参数，从而能够减少不正确反馈信誉值对模型的影响；在此基础上介绍了基于假设检验的反馈检验方法，从而能够进一步抵抗恶意反馈的攻击。文章从理论上证明了 RLM 模型的健壮性。实验结果表明 RLM 信誉模型能够有效地评估信誉估计值和信誉估计方差，与其它信誉模型相比，RLM 能够给出更准确的信誉评估，且具有更高的健壮性。

(3) 设计了一种基于策略和信誉的混合信任管理系统

现有的基于角色的策略语言只能定义[0,1]布尔类型的角色关系，难以支持更细粒度的访问控制，且这种静态的角色管理不能跟踪角色的授权行为，无法抵御角色域内的内鬼攻击。为此本文设计了一个基于角色策略和信誉的混合信任管理系统 RTE (Role-based Trust Evaluation)。RTE 的信任策略语言通过信誉值参数支持信誉的管理，系统的信誉值计算包括信任经验和信任推荐，能够实现资源的细粒度访问控制。另外，RTE 通过定义信任合成算子，能够支持信誉值的网络传递和计算，进而可以根据角色的跟踪记录实现角色授权的动态管理，抗击角色域内的恶意行为。文章给出了 RTE 策略语言的语法和推演规则，介绍了 RTE 系统的信任值计算，并给出了一个 RTE 系统进行混合信任管理的示例。

(4) 提出一种面向可靠性信任的信誉模型及工作流优化算法

当前基于信任的应用大多是基于一次交互的简单应用。对于一个由多个作业顺序或并序组成的网络作业流，评估系统资源的可靠性并在此基础上为其提供面向可靠性信任的资源调度正变得日益重要。现有的大多数用于评估资源可靠性的信誉模型没有考虑作业的运行时间，无法精确评估作业的可靠性；此外多数工作流调度算法采用基于列表的启发式，没有采用能给出更好调度方案的遗传算法。本文提出一种可靠性驱动的 RD (reliability-driven)信誉模型，利用资源的失败率来定义信誉，能够直接被用来评估作业的可靠性。基于 RD 信誉，提出了一种前瞻的遗传算法 LAGA（look-ahead genetic algorithm）同时优化工作流的时间和可靠性信任。LAGA 采出一种全新的演化和评估机制：遗传算子只演化一个调度方案的作业资源映射，而调度的作业顺序由算法的评估步骤采用我们提出的 max-min 策略决定，该策略是第一个能在遗传算法中运行的两阶段工作流启发式。实验结果表明 RD 信誉能够给出更准确的信誉和作业可靠性评估；LAGA 能够给出比现有基于列表启发式更好的工作流调度方案，且比传统的遗传算法有更好的收敛性。

关键词： 信任管理，策略，信任证书，信誉，健壮性，动态角色，可靠性，流调度

ABSTRACT

The internet based information system is becoming increasingly important for the human society. The e-business, e-government and e-society applications are widely deployed on the internet, and various new computing techniques such as cloud, grid and peer-to-peer computing, have made the internet-based applications more convient and efficient. Althougth the internet can result in great benefit, it also causes huge challenge for both the trust in the human society and the trust between human and machaine. With the unfamiliar client and unknown service quality, the trust risk for an internet application has been increased considerably. Hence, a trustworth internet environment becomes one of the preconditions for further improvement of internet based applications.

According to the requirements of the internet based applications, we classify the trust management system into four trust levels: the ID-based trust, the utinity-oriented trust, the reliability-oriented trust and the robustness-oriented trust. For the trust in the first three levels, we put forward four closely related questions: how to design a powerful policy language for the ID-based trust management, how to get a comprehensive and robust reputation model for a service utility, how to combine the policy-based and reputation-based trust management, and how to apply the trust in a complicated network system to optimize its reliability oriented trust. To solve these four questions, our work in this paper can be briefly introducd as following:

(1) Distributed Proving Oriented Language and Method for Trust Negotiation

Most existing trust policy languages cannot simultaneously support the following important characteristics: distributed trust proving, complicated access control definition and negotiation-related constraints. This paper presents the RTP (role-based trust proving) policy language for distributed trust proving and negotiation. The main contributions of RTP include: through entending the RT language, it can define complicated roles; with the predicate *lsign*, it can define a logic role and support the distributed trust proving; with the predicate *release*, it can protect the policy’s sensitive information; and it can avoid the unrelated credential fetching with the help of the negotiation heuristics. Both the syntax and semantics of RTP are introduced. In addition, we proved the soundness and completeness of RTP through the definatation of its inference rules. Based on RTP language, we design a distributed trust proving and negotiation algorithm, which can carry out an efficient trust construction through local trust negotiation or remote trust proving. We also demonstrate the soundness and completeness of DPN algorithm based on the trust graph. Our experiments show that the DPN algorithm outperforms the traditional trust negotiation in terms of both time and number of credential transfers.

(2) A Comprehensive and Robust General Trust Model for Reputation Evaluation

Reputation-based trust management is becoming increasingly important in distributed systems. Although reputation is a prediction about the trust probability, most existing work cannot assess the reputation prediction variance. Moreover, the summation method is widely used for feedback aggregation, but it is vulnerable to malicious feedbacks and difficult to be protected. This paper presents a general trust model RLM, whose highlight is the use of reputation predication variance to give a more comprehensive and robust reputation evaluation. Concretely, we define the reputation by two attributes: reputation value and reputation prediction variance. For feedback aggregation, we introduce the novel Kalman aggregation method, which can support robust trust evaluation. To make the model robust, we design the Expectation Maximization algorithm to mitigate the influence of a malicious feedback, and further apply the hypothesis test method to resist the malicious feedback. Through theory analysis, we demonstrate the robustness of our design. Our experiments show that RLM model can effectively capture the reputation and its prediction variance. Compared with some other typical trust models, RLM can give a more accurate reputation prediction. Moreover, it has a high robustness under the attack of malicious feedbacks.

(3) A Combined Trust Management with Policies and Reputation Value

Most existing role-based policy languages define trust as a Boolean-role relationship [0,1], which cannot support fine granularity access control. In addition, the current static role management system cannot track the authorized role usage to defend the malicious role behavior. In this paper, we propose a trust management system RTE which can combine the advantages of both the policy-based and reputation-based trust management. The policy language of RTE can support the trust value management through adding the parameter of reputation value. We compute the reputation value by the trust experience and trust recommadetion, so that RTE can give a fine granularity access control. In addition, through defining a trust aggregation operator for the social network, RTE can track the role behavior, hence, it can dynamically manage the role and defend the malicious role behavior. Both the synax and the inference rules of the policy language in RTE are introduced, and we give a demonstrating exaple of how to manage the trust with RTE.

(4) Reliability-oriented Reputation and Its Application in Workflow Optimization

Most existing work used the reputation to make a decision for a simple application with only one transaction involved. For a workflow application, which is composed of many sequential or paremerall tasks, providing a reliable scheduling based on resource reliability evaluation is becoming increasingly important. Most existing reputation models used for reliability evaluation ignore the task runtime influence. Moreover, to optimize makespan and reliability for workflow applications, most existing works use list heuristics rather than genetic algorithms (GAs) which can usually give better

solutions. Hence, in this paper, we propose the reliability-driven (RD) reputation, which is time dependent and can be used to evaluate a task's reliability directly using the exponential failure model. Based on RD reputation, we propose a look-ahead genetic algorithm (LAGA) to optimize both time and reliability for a workflow application. LAGA uses a novel evolution and evaluation mechanism: the evolution operators evolve the task-resource mapping for a scheduling solution, while the solution's task order is determined in the evaluation step using our proposed max-min strategy, which is the first two phase strategy that can work with GAs. The experiments show that the RD reputation can improve the reliability for an application with more accurate reputation, while LAGA can provide better solutions than existing list heuristics and evolve to better solutions more quickly than a traditional genetic algorithm.

Keywords: trust management, policy, credential, reputation, robustness, dynamic role, reliability, workflow scheduling

第一章　绪论

本章首先介绍我们进行互联网环境下信任问题研究的背景，然后针对当前网络应用的信任需求，陈述本文关注的四个信任研究问题，以及我们的解决方案和论文贡献，最后给出全文的组织结构。

1.1 问题背景

随着网络技术和计算机技术的快速融合，基于互联网的应用呈现出爆炸式的发展态势。政府、企业和各科研机构纷纷将自己的应用系统部署到互联网上，以便于信息共享和提高工作效率；而各种新型分布式计算技术也使得基于互联网的应用更加简单高效，典型技术包括云计算、Grid 以及 P2P 技术等。所有这一切都促使互联网成为世界上最大的人造信息系统，并构成了一个分布协同的虚拟信息社会。互联网信息社会在给人们带来方便的同时，也给系统和应用的信任关系带来了巨大的挑战。正如 Peter Steiner 所说“在 Internet 世界里，没人知道你是一条狗”；David Nicol 则说“在 Internet 世界里，每个人都可以说你是一条狗，但没人知道你会不会咬人”[2]。这实际说明了网络世界中的身份信任和行为信任两个问题。因此，如何在互联网环境中，对用户的身份信任和行为信任进行有效管理成为互联网应用拓展的重要基础。而这一挑战已经超出了传统网络安全处理的范畴，为此，人们提出了各种信任管理技术。本节首先介绍分布协同的互联网计算环境的基本特点；然后阐述新的计算环境下一些典型的信任挑战以及技术难点；最后论述目前被广泛关注的信任管理技术。

1.1.1 互联网与信任

作为计算机技术与通信技术融合的产物，互联网正从传统的计算机通信平台演变为无处不在的分布式网络计算平台。过去，人们努力使计算机具有联网功能，现在则是强调网络具有计算能力。尤其随着各种移动个人计算 PC（Personal Computing）设备的广泛部署和应用，互联网已成为人们生活工作的必备工具和载体。人们将更多的带有隐私和安全要求的个人信息交由互联网存储或处理，如公共邮件系统以及各种社会网络平台（Facebook，校内网等）；原先服务于单个组织、政府或科研机构的应用系统正快速转变为面向互联网的开放式系统，电子商务、电子政务以及电子科学（e-science）等应运而生，以便能更好地处理和共享服务。

受到需求的驱动，工业界和学术界提出了多种网络计算技术（如网格计算、

对等计算和云计算）。随着面向服务体系架构 SOA（Service Oriented Architecture）和软件即服务 SaaS（Software as a Service）等理念的日益成熟，软件系统应用模式凸显出网络化和服务化的趋势。无论网格计算，对等计算还是云计算均是从不同侧面对面向网络的分布计算模式进行积极探索，它们的共同之处是如何基于开放、动态的网络计算环境，实现可信、协同的资源(包括计算资源、数据资源、软件资源以及服务资源等)共享和有效利用[16]。由于互联网上各类分布异构资源缺乏有效的安全控制和组织管理机制，可协同、可管理和可信任成为互联网应用系统的三个基本问题。具体地，可协同性问题是指如何跨自治域进行资源共享和协同工作；可管理性问题是指如何将异构、庞大的网络资源进行有效地聚合与管理；可信任性问题是指面对大量不可控和缺乏信任基础的服务资源，如何保证共享和协同资源之间可靠和相互信赖的信任关系[16]。

可信任作为基于互联网应用系统的基本要求，它在新的计算环境和应用模式中面临诸多新的挑战。如何在互联网环境下对信任进行有效管理，以适应计算环境和应用模式发展的需要，已经成为当前的热点问题。互联网环境下，信任管理的需求主要根源于互联网资源的一些自然特性。

1． 分布无中心性

互联网作为一种分布式计算系统，用户可以从任何地方简单动态地访问并处理数据，这给安全管理带来了极大的挑战。为了管理分布的用户访问，一个重要的解决方案就是通过安全架构对用户进行等级划分。在集中式管理系统中，解决该问题只需维护一个安全服务器就可以完成用户安全等级的分配和认证问题。但是在分布式计算环境中，由于存在多个安全域，安全等级的管理机制与集中环境相比有着更大的复杂性。需要一个有效的安全框架通过标准的安全协议来屏蔽不同安全域之间的差异，同时保证不同的安全域仍可以有效地保护内部的各种资源。

分布式安全框架及协议首先需要一个有效的用户身份管理机制。传统的解决方案是把各个安全域中的用户映射到一个统一的环境中。在网络规模和用户数量有限的情况下，该机制还可以对用户身份进行静态分配。但是，随着网络用户数量的不断增加，这种静态分配用户身份的可操作性就变得越来越低。而系统的无安全控制中心特性，使得难以形成集中的资源授权和信任关系定义。这需要多域联盟、推荐及委托等信任机制的支持，使得信任的网络传递管理成为互联网环境下信任管理的一个重要问题。

2． 自治性

自治性是指互联网资源具有局部自治、自主决策的特性，用户节点作为资源的所有者，基于“自愿参与、自主协同”的原则自发构造系统的行为。例如，互

联网上很多资源是面向特定组织和个人的，资源提供方既可以根据资源请求方的要求向它提供资源，也可以拒绝提供资源。在 P2P 和 Grid 为代表的互联网自治系统中，节点作为用户的代理，一个基本的特征是其在参与系统过程中的自主性。在自治系统中，无论节点的加入、退出，还是系统内部节点间的相互通信和服务，都是由节点自身发起和唯一决定的，节点的行为不受系统本身的直接控制。

互联网资源的自治性使得系统中的节点行为只取决于其自身的利益或准则，无集中控制。正因为如此，相对于传统的分布式计算系统，基于互联网的应用系统在运行特征和运行轨迹上通常都表现出更大的不确定性和动态特征[21]，资源的行为更加无法控制，而这对资源的信任问题带来无比挑战。

3． 协同性

互联网系统中，系统应用通常都会涉及到多自治域间的服务调用和组合。为此，一个首先需要解决的问题是如何在陌生的来源于不同自治域的实体间促成协同活动。所谓协同是指多个资源为完成共同任务而进行的交互、同步和计算的过程。由于互联网资源具有分布自治性，各个管理域在互联网环境下广泛分布，可能跨越多个国家；每个管理域都有各自的资源访问控制规则以及协商策略，不同的管理域会使用不同的规则。这些都使得如何在互联网资源间协同地建立信任成为跨域协作的瓶颈性难题。它意味着需要在大量的机器上，根据不同自治域的各种要求，同时同步一个统一的信任策略集合，并根据这个策略集合，对来自不同自治域的用户进行统一管理。

4． 异构动态性

异构性是指互联网资源属性存在广泛差异的特性。这些差异增加了资源统一建模和管理的难度。动态性是指互联网资源规模不断膨胀、资源关联关系不断变化的动态特性。互联网是一个不断成长的开放系统，其覆盖地域不断扩大，大量的分布异构资源不断更新与扩展。随着互联网应用的扩展与深入，资源及其关联关系也在不断动态演化，相应的资源特征信息也必将不断的扩充和变化。其次，系统的动态演化特性，体现为主体的加入退出行为频繁，导致协作环境及信任关系随需而变，对有效的信任管理需求增强。

1.1.2 信任关系的挑战

互联网计算环境的分布无中心性、自治协同性和异构动态性对信任管理提出了新的要求。下面通过分析三个典型的信任关系挑战问题，进一步具体介绍互联网环境下信任管理需解决的问题。

1. 跨域访问控制的信任问题

跨安全域的资源访问控制是分布协同系统首先需要解决的问题，下面我们引用文献[62]中的一个例子简要说明跨域进行访问控制信任建立的问题。

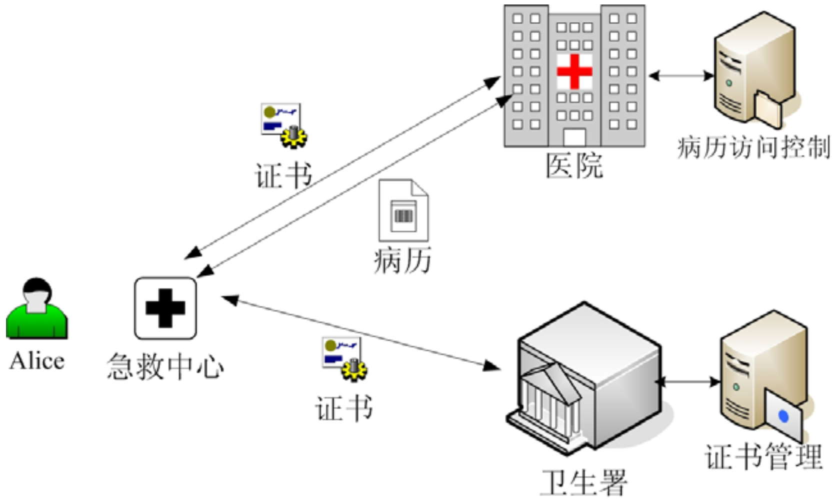

图 1.1 医疗急救中的访问控制信任

实例 1(医疗紧急救助):

如图 1.1 所示, 在对 Alice 实施的一次医疗急救中，急救中心 FirstAid 需要向 Alice 曾就医的医院 Hospital 请求访问其电子病历 R，然而，R 涉及到 Alice 的个人隐私信息，属于敏感资源，所以，Hospital 制定了相应的保护资源 R 的访问控制策略：只有 Alice 本人及急救中心才能调阅 R。在协议消息交互过程中，各方会根据其独立的协商策略披露相应的消息项，如，FirstAid 只要提交当地卫生署为其签发的急救中心信任证书，就能快速地访问到病历 R。因此，建立跨安全域之间的信任通常面临着如下几项本质问题：

(1) 当隶属类似于实例 1 中的机构 FirstAid 和 Hospital 两个独立安全域的陌生主体进行资源访问时，如何提供一种有效的方法和机制，以动态地建立两者的信任关系？

(2) 当开放网络中的协商主体在维护其自治性和隐私性时，需要什么样的访问控制策略和信任证书(如实例 1 中 Hospital 制定的访问控制策略和 FirstAid 拥有的信任证书)?

(3) 对资源的访问控制结论，不再是单纯的 Yes 或 No，需要根据各自的协商策略给出相应的提议，以支持进一步的协商：既要实施信息保护，又要达成联合协作。因此，如何建立协商策略机制以兼顾二者的要求?

(4) 此外，信任的建立将依赖于一套完整的协议，例如，在实例 1 中体现为机构 FirstAid 和 Hospital 的消息交互过程。

2． 不可信甚至危害性服务问题

在分布协同的计算环境中，没有集中控制的服务器。每个参与节点可能具有不同的动机，并且不同的节点具有不同的能力，由此导致系统可能存在不可靠甚至危害性的服务资源。不可靠服务的具体表现是提供不真实（例如虚假的文件下载）或与所声明服务质量不符的服务，其目的通常是降低系统整体服务的可靠程度，使得用户难以获得真实可信的服务，从而降低或破坏其对于系统本身的信任，影响系统的健康运行以及进一步的扩充和发展。仅就 P2P 文件共享系统而言，以不可信文件为代表的不可靠甚至危害性服务问题已成为影响其整体应用价值的关键问题之一[25]。2005 年学者对 KazAa 中最流行的 100 个文件的统计发现，其中不可信文件的比例平均超过了 50%以上[26]；而 Gnutella 系统内部不可信文件的泛滥已使之作为一个具有应用价值系统的地位基本丧失。

3． 网络节点自私性问题

分布自治的计算环境存在另一个主要问题，即节点的自私问题。系统的每个节点只追求节点本身的利益最大化，而忽视整个系统的性能。P2P 网络中，典型的自私问题包括搭便车(Free-Rider)和资源无节制使用(Tragedy of the Commons)问题。搭便车特指那些加入系统后总是使用其它节点共享的资源，却很少或从不提供资源共享的节点。搭便车现象的泛滥，直接导致了以 Gnutella 为代表的很多 P2P 文件共享系统内部可用资源的不足。资源无节制使用问题是指网络带宽作为一种非排他占有的公共资源，被各种 P2P 节点无节制使用的现象。研究表明，P2P 节点无节制使用网络带宽的行为已经成为影响互联网应用价值[36]的重要问题。节点的自私问题主要源于节点对其它节点成为“优秀公民”的不信任，以及系统缺乏监督和激励机制。如何建立节点对其它节点以及系统惩罚和激励机制的信任，成为一个分布自治系统能够吸引资源，稳定发展的基本要求。

以上三个问题基本反映了互联网环境所面临的信任建立问题。第一个挑战涉及如何根据用户身份的不同进行资源访问控制，我们将它称为基于身份的信任问题。后面两个问题不同于传统的系统安全性[39, 40]问题，它们是自治节点参与系统过程中产生的对系统造成不良影响的行为，我们称之为基于行为的信任问题。信任是人类社会的重要基石之一，在社会科学、技术科学、商业等诸多领域中都发挥着关键作用。它长期以来被认为是主观的、不精确的、不可靠的、甚至不可信的，缺乏科学的系统的研究，尤其是缺乏形式化的定量研究。随着互联网的出现和发展，人们对信任进行形式化研究的要求变得越来越迫切。作为当前计算机科学中仍处于探索阶段的领域，信任管理技术的研究面临着诸多挑战。

1.1.3 信任管理的提出

在互联网计算环境中，为了应对各种新的信任建立挑战，信任管理逐渐成为一个重要的安全技术。下面从技术演化的角度，简要概述基于身份的访问控制信任管理和基于信誉评价的信任管理技术。

1．基于身份访问控制的信任管理

基于身份访问控制的信任管理主要通过身份进行资源访问信任关系的管理。计算机安全是一个具有 30 多年历史的研究领域，以认证（authentication）、授权（authorization）和审计（audit）为基础的 AAA 安全是实现系统安全性的主要安全技术[25]。认证是访问控制的基础，是分布系统安全的重要基础设施。授权管理是一类具有服务质量保证和管理性质的技术。审计是对授权行为的监控，是计算机安全的必备要素。基于 AAA 技术的资源访问控制是计算机安全的核心内容，其研究范围涵盖访问控制策略、访问控制模型和安全管理等诸多研究分支，核心任务是确保资源访问限于具有合法权限的主体，同时保证具有合法权限的主体能够访问到相应资源[26]。

计算机系统形态经历了集中式主机系统、客户/服务器系统和基于互联网的系统三个主要阶段[129]。相应的，计算机资源访问控制也从传统的访问控制、分布式访问控制、逐步发展到面向互联网的信任管理技术阶段。早期的访问控制技术主要关注多用户主机系统的资源共享问题。针对操作系统和军事信息系统的不同安全需求，人们提出了自主访问控制（Discretionary Access Control, DAC）[27]和强制访问控制（Mandatory Access Control, MAC）[29]两类具有代表性的访问控制策略。由于 DAC 和 MAC 在解决商业组织安全问题时存在局限性，自 1992 年起，人们提出多种基于角色的访问控制（Role Based Access Control, RBAC）模型[33,34,35,36]。

早期的访问控制技术主要采取集中方式进行管理，在可伸缩性和灵活性等方面存在局限性，由此人们提出分布式访问控制技术。它的一个重要特点是将传统访问控制技术与密码技术结合，访问能力和主体身份的分布化是这个阶段研究工作的主要特点。上世纪 80 年代末至 90 年代初，出现了大量研究多级访问请求、多级认证和特权委派的研究工作[64,65,66]，这类系统的主要目标是实现信任链中的分布式身份推演和委派机制。

分布式访问控制技术适用于以身份集中管理方式进行的分布式应用系统。但是基于 PKI 的安全技术依赖于全局命名体系，在实践中面临诸多难题[13]；此外，在域间授权活动中，主体身份对服务方可能是陌生的，身份不足以作为授权的依据。为此学者们提出信任管理（Trust Management）的概念[18]，它被定义为：“采用一种统一的方法描述和解释安全策略、安全凭证和用于直接授权关键性安全操作的信任关系”[9]。信任管理系统把授权决策转换为一种满足性验证（Proof Of

Compliance）问题："凭证集 C 是否能证明操作 A 满足本地策略 P"，其中 C 是请求方和授权方收集的凭证，A 是请求方希望执行的操作，P 是授权方的本地策略。信任管理是一种面向分治系统的访问控制方法，基于委派实现灵活可伸缩的授权，能够有效解决陌生人授权问题。

2．基于行为信誉评价的信任管理

访问控制和授权管理关注的是用户的身份可信问题，信任管理还需解决用户的行为可信问题。用户的行为可信是指终端用户的行为是否可以评估、可预期、可管理、对网络设备和数据是否会造成破坏或毁坏。传统的安全机制可以解决用户的身份信任问题，但并不能处理用户的行为信任问题。例如在数字化电子资源的订购方面，大学生通过合法可信的身份(学校的 IP 地址或帐号)可以登录到学校定购的数字资源服务器上，但他的行为却有可能是不可信的。例如，某些学生在校内使用网络下载工具大批量下载学校购买的电子资源，再通过私设代理服务器牟取非法所得等，即用户的身份是可信的，但用户的行为不一定可信。

用户行为信任的研究最初来源于社会学、经济学和心理学等领域。在计算科学领域，信誉作为对用户行为信任的管理技术，它在很多研究方向中得到关注。在文献[8]中，Jøsang 是这样定义信誉的：信誉是关于某个人或某件事的特征或者立场的大众观点。由于信誉系统能够提供信任信息的聚合、过滤以及排序的功能，它在很多领域都被广泛应用，如分布式系统和应用[85,86,87,88,95]，多 Agent 系统[90,91]，基于 Web 的服务[92,93] 和推荐系统[84,106,110]等。

1.2 本文研究的信任问题

针对分布协同的互联网环境，本课题把信任管理按可信系统的需求层次进行划分，并对部分重要且相关的信任问题进行研究。如图 1.2 所示，我们根据互联网系统的需求层次将信任分为四个层次：

1．基于身份的信任（ID Trust）。可信系统首先需要保证服务资源只能被使用真实身份且已被授权的用户访问，即保证合法的用户和合法的访问。该层次信任主要用于解决系统的资源访问控制，因此也称系统访问控制的信任。

2．服务属性信任（Utility Trust）。在身份信任的基础上，即当用户使用适当的身份获取具有某些属性的服务后，可信系统还必须确保该用户正确使用该服务属性，即保证用户的服务操作行为合法。例如，在云存储计算环境中，如果某些合法的存储服务商被允许出售保密数据存储服务，那么可信存储环境就必须确保该服务商能够提供保密存储的服务属性。服务属性信任主要依靠服务的信誉来管理，因此也称为基于行为信誉的信任。

3．面向可靠性的信任（Reliability Trust）。在服务属性信任的基础上，可信系统还必须保证某种属性服务的可靠性，即用户提交的任务能够在保证服务属性的前提下，顺利完成。例如，在上例中，系统需保证用户的保密数据存储不会在用户撤销之前，因物理故障等原因而丢失。

4．面向健壮性的信任（Robustness Trust）。在面向可靠性信任的基础上，可信系统还可进一步提供面向健壮性的信任。也就是说，即使系统发生了一些故障，系统仍可以完成某些属性的服务。

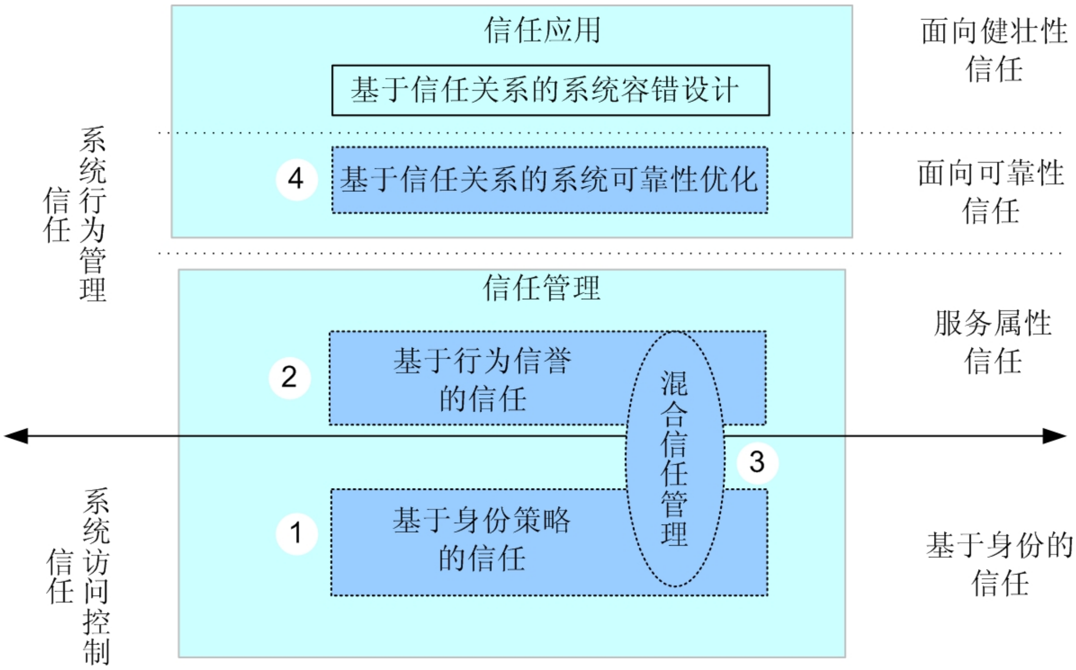

图 1.2 互联网系统的信任层次划分

上面四种信任关系逐层叠进，最终完成可信系统的建设。基于身份的信任作为系统访问控制的信任，是信任管理的基础；服务属性信任、面向可靠性和健壮性的信任都是对系统行为的管理，因此它们构成了系统行为信任的管理。另外，基于身份和信誉的信任是对信任概念和内涵本身的管理；而面向可靠性和健壮性的信任乃是利用身份和信誉信任来提高系统的性能，因此它们属于对信任本体的进一步应用。本文将对紧密相关的前三个层次的信任进行研究，如图 1.2 中蓝色标注的四个部分所示，本课题研究的四个问题如下：

1.2.1 基于身份策略的信任描述

身份策略即是基于身份的访问控制策略。在基于身份策略的信任管理中，如何对分布式环境中跨安全域的资源进行访问授权管理成为当前的研究热点。其中一个主要研究方向为自动信任协商(ATN, Automated Trust Negotiation)。与传统基

于访问控制列表的方法相比，它的特点是通过信任证书和访问控制策略的交互披露，资源的请求方和提供方自动地建立信任关系[51]。

目前对 ATN 的研究主要有两方面：信任证书的分布式放置收集方法以及策略语言的形式化描述分析。信任证书分布式放置收集方法包括传统的信任协商和分布式证明两种方法[53]，Bauer[54]分析认为分布式证明方法比传统信任协商方法更高效并能克服后者的一些缺点，如信任求证节点负载过大。ATN 策略语言的内容可分为资源访问控制策略和信任协商策略，一般情况下策略语言将两者分离并分别进行描述[52]。策略语言的资源访问控制策略利用角色对权限的授权和委托信息进行定义，角色的描述能力越强对应用的支持就越大。协商策略需提供对信任证书敏感信息的保护并避免信任证书的盲目搜索。综合自动信任协商系统的上述特点，一个好的信任管理策略语言必须具有强大的资源访问控制描述能力，能够保护信任证书敏感信息，避免信任证书的盲目搜索并能支持信任分布式证明。

目前一些研究工作都只是对我们目标的某个方面提出解决办法，如[54]通过定义 say 谓词提出了一个信任分布式证明算法，但该算法不支持复杂的资源访问控制和信任证书信息保护；N.Li 在文献[55]中提出一种资源访问控制策略 RT(Role-based Trust-management)语言，支持参数化和连接两种复杂角色，但不能支持敏感信息保护；J.Li[56]在 RT 语言的基础上通过加密信任证书技术支持敏感信息保护，但这两者都不能支持信任的分布式证明方法。以上语言和方法之所以不能对信任分布式证明方法以及复杂的资源访问和信任协商策略提供全面支持，主要是缺少一个功能全面的策略描述语言。

1.2.2 基于行为信誉的信任模型

建立健壮的信誉机制，实现可靠的信任评价，是基于信誉的信任管理系统成功的基础。信誉是从历史行为记录中得到的一个关于信任概率的统计值。通常情况下，信誉值可以通过两种途径进行计算：根据评估者自身经历的直接信任和根据信誉反馈的间接信任[8]。为了能够计算直接信任，研究者提出了很多信誉模型，包括简单平均模型[83]、Bayesian 模型[94]以及基于证据的信念模型[90,91]等，他们将信任量化成一个或几个确定性的估计值。虽然信誉本质上是一个概率的估计值，但现有的这些信任模型大多忽略了概率估计值的另一个重要属性：概率估计方差。信誉（可信概率）估计方差可以评估信誉估计值和真实信誉值之间的偏差，它在信誉系统中可以起到很重要的作用。例如，一个服务资源真实的事务成功率是 90%，一个服务消费者 A 与该服务资源进行了少量的事务交互，并根据自己的记录将该资源的信誉值确定为 70%。现有的信誉模型将不能支持评估者 A 对该信誉估计值的偏差进行评估，这将带来两个后果：(1)评估者不能确定自己给出的信誉

估计值的准确性，因而也就不能确定该多大程度的依靠该信誉评估做出决策。(2) 当评估者 A 将该信誉估计值作为反馈发送给其它信誉评估者 B 时，A 无法通告 B 应如何聚合该信誉反馈从而能得到更准确的信誉评估。

为了能够聚合信誉反馈，一个很重要的问题是如何处理恶意推荐问题。恶意节点对具有良好行为的节点进行诋毁提交负面的评价，对恶意行为的节点进行夸大提交正面评价，这将严重扰乱系统的资源配置，进而降低系统的性能。目前的大多数工作采用相加的聚合方法[85,86,88,108]，但是该方法很容易被恶意信誉反馈攻击[96]。为了解决这个问题，很多研究工作通过计算信誉反馈的可信度来检测恶意反馈[96,103,108]。这些反馈可信度计算方法一般都假设评估者知道整个信誉系统的一些全局信任知识[85,96,108]，或者需要一些人为设定的感性参数[89,91]，这种强假设条件在实际应用中可能是无法实现的。这些基于相加方法的可信度检测技术之所以难以应用，原因在于相加的聚合方法缺乏对健壮性推测技术的支持。虽然相加的方法容易进行反馈聚合，但这种感性的聚合方法没有统计推测理论的基础，容易被恶意节点控制。

综合上面的分析，为了能够得到更全面和健壮的信誉评估，我们需要研究如下问题：1. 如何在信任模型中考虑信任评估的方差，并同时给出信誉值及其评估偏差的估计，从而给出更全面地信誉评估； 2. 如何用模型预测的方法来聚合信任反馈，从而能够避免使用感性的相加聚合方法，以支持对恶意反馈的抵抗；3. 如何能够自主的对信任模型进行健壮性的较准，从而能够在不需要全局信任知识及人为参与的情况下，模型能够自主准确地抵御恶意推荐攻击。

1.2.3 基于策略和信誉的混合信任管理

基于策略的信任是一种理性信任，可以用布尔值表示。基于信誉的信任是一种感性信，是主体对客体特定行为的主观可能性预期，取决于经验并随着客体行为的结果变化而不断修正。基于策略和基于信誉的两种信任关系具有很强的互补性。一方面在计算机的安全技术发展中，认证和授权、访问控制等都可以看作是基于策略的信任。这种理性信任关系在安全控制领域一直发挥着重要的作用，它们是感性信任的基础。另一方面，基于信誉的信任关系是一种感性信任关系，更符合人类对信任基本性质的认识，并弥补理性信任的不足。感性信任试图通过模拟人类社会中的交互和信任关系，在计算机世界中建立起一种量化的互信关系，从而可以辅助决策制定。另外，感性信任关系还可以表示信任关系的历史、现状和将来的发展趋势，可以应用于某些复杂领域的信任问题。因此，感性信任关系在某种程度上弥补了理性信任的不足。

目前的大多数的信任管理都将两者分离开，单独进行信任关系处理。对于理

性信任模型而言，它们将信任定义为一个客观的逻辑关系，可以抽象为 0 或 1。但在实际的应用中，大多数情况下用户希望能够实现更细粒度的信任管理，比如该多大程度地相信一个经济分析师的投资推荐，而不仅仅是相信或者不相信。另外，目前大多数基于角色的凭证管理技术是一种静态的信任管理技术，它不能跟踪角色的授权行为并动态管理角色关系，这将导致角色域内的用户行为无法被跟踪控制，并为角色域内的恶意行为提供了条件。

因此如何结合基于策略和基于信誉的两种信任，设计出一种混合的信任管理机制，成为我们的目标。为此主要的研究问题包括：1．如何在基于角色的凭证管理中引入信誉属性，从而拓展普通意义上的角色描述能力；2．如何利用带信誉的角色关系进行更细粒度的授权管理。3. 如何通过角色的信誉值，自动探测角色域内的恶意行为，设计出健壮的角色管理技术，增强系统抵御内鬼攻击的能力。

1.2.4 面向可靠性的信任模型及其优化

目前对信任本身的研究工作很多，但对基于信任关系的应用研究相对较少。例如，学者们在 P2P 及社会网络中提出了很多信誉系统，但只有很少的研究工作[87]采用信誉来增强复杂系统的可靠性和健壮性，且大多数基于信誉的应用都比较简单如 ebay 等，都是一次事件交互应用。另外，目前的大多数基于信誉的信任模型难以支持复杂应用系统的可靠性量化评估。例如为了评估资源的可靠性，很多分布式和多 agent 应用[8,91,86,109]采用信誉系统来跟踪资源的历史行为。然而这些信誉系统存在两个问题：1.大多数信誉模型[8,108,121,86]将资源的信誉定义为该资源成功完成作业的概率，在信誉计算过程中，他们忽略了作业运行时间（大小）的影响。2.在大多数基于信誉的资源调度系统中，运行在某个资源之上的所有作业具有相同的可靠性（成功率），它被定义为资源的信誉[87,121]，也忽略了作业运行时间的影响。

针对上面提到的阻碍信任关系应用的两个问题，我们将研究适用于复杂应用可靠性评估的信誉模型，并研究利用该信誉对网络作业流的可靠性进行优化的算法和机制。具体的研究问题包括：1.如何对分布式系统中资源的运行失败进行模拟，并提出一个基于可靠性的信誉模型，它能够实时的跟踪资源的失败率，并能提供作业可靠性的定量度量；2.如何利用我们的信誉评估结果，提出适用于复杂作业流的启发式规则；3.如何根据资源的信誉评估结果，对系统的资源调度和管理进行优化以便达到最佳的稳定性和可靠性。

1.3 论文的主要贡献

针对上小节中我们提出的四个有关信任的研究问题，本文将分别提出我们的解决方案。论文的主要贡献如下：

1．提出一种支持分布式证明和协商的信任策略语言和方法

现有的大多数信任策略描述语言无法对信任协商的需求提供全面的支持。为此我们的目标是设计出一个功能全面的信任管理策略语言，它具有强大的资源访问控制描述能力，能够保护信任证书敏感信息，避免信任证书的盲目搜索并能支持信任分布式证明。

本文提出一种面向信任分布式证明和协商的策略语言 RTP(Role-based Trust Proving)。RTP 语言的主要特点如下：1）对 RT 语言进行改进和功能拓展，从而使 RTP 语言的访问控制策略能够延续 RT 语言描述能力强的优点，可以定义复杂的角色如连接角色和带参数的角色。2）语言中增加 lsign 语法，可以定义逻辑推导角色，且该新增类型角色是对传统角色定义的放松，能够支持信任分布式证明。3）语言的信任协商策略增加 release 谓词，从而可以限制信任证书信息的传播，提供对信任证书保护技术的支持。4）信任协商策略中增加 prove 和 find 谓词，可以定义信任协商启发式规则，从而避免信任证书的盲目搜索。文章详细介绍了 RTP 语言的语法构成，定义了 RTP 语言的推理证明规则，给出了语言的语义解释并证明了语言的可靠性和完全性。

为了体现 RTP 语言的全面功能，本文还提出一个基于 RTP 语言的信任分布式证明协商算法 DPN(distributed proving and negotiation)。DPN 算法通过本地信任协商和远程信任证明，可以高效地完成信任分布式证明任务。另外通过信任证书限制策略和启发式规则，算法可以有效地保护信任证书敏感信息，并避免信任证书盲目搜索。文章通过信任图的概念分析了算法的正确性和完整性。我们的实验表明，跟传统的信任协商方法相比，DPN 算法能够有效地减少信任建立时间和交互次数。

2．设计一个全面健壮的通用信誉模型

尽管信誉是信任概率的一个统计估计值，但现有的大多数信任模型没有考虑信誉估计偏差，因此无法估计信誉评估本身的准确性。另外大多数现有工作采用相加的方法聚合信誉反馈，该感性方法容易遭受恶意反馈的攻击，且很难被拓展以增强健壮性。

为了能够得到一个全面和健壮的信誉评估，本文介绍了一种健壮的线性马尔科夫 RLM（Robust Linear Markov）信誉模型。该模型的主要贡献包括：

(1) RLM 模型将信誉评估表示成由信誉估计值和信誉估计方差组成的二维元组，从而能够得到更全面地信誉评估。采用线性自回归方程定义信誉状态空间的

演化，从而构成了一个隐马尔科夫过程。

(2) 模型采用卡尔曼滤波方法聚合信誉反馈，通过反馈噪音方差这个模型参数，卡尔曼聚合方法可以控制一个不正确反馈信誉值对模型的影响。该性质能够支持模型进一步采用健壮性的统计推测技术以抵御恶意反馈攻击。

(3) 设计了一个健壮性的模型校准方法。为了计算模型中的动态参数，模型首先采用 EM (Expectation Maximization)参数估计方法，它能自动产生适当的反馈噪音方差从而减轻一个恶意反馈信誉值的影响；在 EM 基础上，模型进一步采用基于假设检验的反馈检测方法，可以抵抗恶意反馈的攻击。文章通过理论分析，证明了模型的健壮性。

RLM 信誉模型及卡尔曼反馈聚合方法完全基于统计推测理论。据我们所知，RLM 是第一个可以评估信誉估计方差的信誉模型，它能够给出更全面准确的信誉评估。我们的卡尔曼反馈聚合方法以及模型校准方法能够抵御恶意反馈的攻击，并能自主地通过局部信任知识计算模型参数，无需人为设定参数。另外，文章同时给出理论分析和实验结果，证明 RLM 信誉模型的有效性、准确性和健壮性。

3．设计一种基于策略和信誉的混合信任管理系统

基于策略（理性）和基于信誉（感性）的两种信任关系具有很强的互补性。理性信任的静态安全机制是感性信任的基础，而感性信任的动态变化性则可以弥补理性信任的不足。然而，目前大多数信任管理系统将两者分离，单独进行信任关系处理，它们无法提供全面的信任评估功能。因此我们的目标是结合基于策略和基于信誉的两种信任，设计出一种混合的信任管理机制，它能够支持如下特性：1)信任管理语言能够支持带信誉属性的角色定义；2)支持细粒度的资源访问控制；3)信任管理语言能够支持信誉的网络计算；4)实现角色的动态管理，通过跟踪角色的授权行为，抵御角色域内恶意行为。

针对上文提出的目标，本文介绍了一个基于角色策略和信誉的混合信任管理系统 RTE (Role-based Trust Evaluation)。RTE 的信任策略语言通过在基于角色的信任关系语言中增加信誉值参数，从而能够在信任策略语言中支持信誉的管理。RTE 的信誉值计算包括信任经验和信任推荐，能够实现资源的细粒度访问控制授权。另外，RTE 的策略语言通过定义信任合成算子，能够支持信誉值的网络传递和计算，进而可以根据角色的跟踪记录，动态管理角色授权，抗击角色域内的恶意行为。文章给出了 RTE 策略语言的语法和推演规则，介绍了 RTE 系统的信任值计算，并给出了一个 RTE 系统进行混合信任管理的示例。

4．提出一种面向可靠性的信誉模型及其在工作流优化中的应用

当前基于信任的应用都比较简单，大多是一次交互作业，缺少大规模的应用

示范。另外目前的信任模型难以支持复杂应用系统的可靠性信任量化评估。为此，本文的目标是对信任的关键网络应用进行研究。由于一个网络作业流是多个作业的顺序或者并序的结合，它具有大规模网络服务应用的典型特点。因此，本文将通过对网格作业流的分析，提出一种基于信任的面向可靠性的作业流的调度系统。另外，由于目前的基于信誉的信任系统大多忽略了事务交互的时间因素，无法支持复杂作业流的可靠性评估。因此，本文还将研究如何量化信任以便在作业流应用中对作业的可靠性进行评估。

针对上文提出的关键信任应用的研究目标，本文首先提出一种可靠性驱动 RD (Reliability-Driven)资源信誉模型，RD 信誉模型采用与时间相关的作业失败率来定义资源的信誉，从而在信誉模型中考虑作业运行时间的影响。此外，文章给出的 RD 信誉算法能够实时地跟踪信誉的变化，并可被用来直接评估作业的可靠性。

基于 RD 信誉，文章提出了一种前瞻的工作流遗传调度算法 LAGA (Look-Ahead Genetic Algorithm)，它可对工作流的时间和可靠性信任同时进行智能的优化。LAGA 具有两个重要特点：1. 基于本文提出的资源优先级启发式，LAGA 的遗传算子可智能地变异调度方案，避免传统的随机变异带来的问题；2. 使用一种新颖的演化和评估机制：遗传算子（交叉和变异）只负责演化调度方案的作业资源映射，而调度方案的作业顺序由算法的评估步骤采用本文提出的 max-min 策略进行智能决策。本文提出的 max-min 策略是第一个能在遗传算法中运行的两阶段工作流启发式。依赖该策略，LAGA 能够避免 BGA 算法中的无效调度方案问题[137]，且能够通过智能的演化方法达到更好的收敛性。

1.4 本文组织结构

本文共分为七章，第一章为绪论，第二章介绍相关工作，我们提出的四个问题的解决方案将在第三、四、五和六章中分别介绍，文章最后一章是总结。各章节的主要内容安排如下：

第一章为绪论，首先介绍课题的背景：分析了分布协同的互联网环境对信任的需求，介绍了互联网环境下信任建立的挑战，并得出信任管理概念的由来。然后详细阐述了本文关注的三个层次的四个递进信任管理问题。针对四个信任问题，简要介绍了本文提出的解决方案以及理论和实验贡献。本章最后是论文的组织情况。

第二章对信任管理的相关概念和工作进行了介绍。陈述了信任的定义，以及它的基本性质和分类方法；介绍了网络系统信任的定义和内涵；分析了信任管理的需求及其种类。针对基于身份策略的信任管理，介绍了策略信任证书的描述语

言和分布式管理方法，给出了典型示例。针对基于信誉的信任管理，总结了信任信息的存储和收集方法，介绍了信誉的多种数学模型。

第三章提出一种支持分布式证明和协商的信任策略语言RTP。首先描述了RTP语言的语法结构和一个知识库示例，定义了语言的推理证明规则。然后介绍了RTP语言的语义解释，并证明了语言的可靠性和完全性。在RTP语言的基础上，本章提出一个信任分布式证明协商算法DPN，并分析了该算法的正确性和完整性。最后本章通过实验说明了RTP语言及DPN算法给信任协商带来的性能提升。

第四章设计了一个全面健壮的通用信誉模型RLM。首先描述了RLM信誉模型的各数学要素以及它们之间的关系。然后介绍了模型的卡尔曼信誉反馈聚合方法，并证明了它对健壮性信誉评估技术的支持。为了使模型能够抵御恶意反馈的攻击，本章描述了EM参数自主校准算法以及基于假设检验的反馈检测方法，并证明了模型的健壮性。最后，本章给出了充分的实验过程，证明RLM模型的有效性、准确性和健壮性。

第五章陈述了一种支持信誉值的基于策略语言的混合信任管理系统RTE。首先提出带参数信誉值的基于角色的信任关系语言，介绍了语言的语法和语义，并给出了带信誉计算的推导规则。然后介绍了RTE系统中的信任计算方法，包括信任经验计算和信任推荐计算以及两者的合成方法。最后，本章介绍一个基于RTE系统进行的资源访问控制实例。

第六章提出一种可靠性驱动的RD资源信誉模型，并在此基础上设计一种工作流调度算法。首先介绍采用作业失败率定义的RD信誉模型，并给出能够实时跟踪信誉变化的RD信誉算法。基于RD信誉，定义了面向可靠性和时间的工作流调度问题。然后，本章提出了前瞻的遗传调度算法LAGA，介绍了算法的各个操作步骤，并分析了算法的时间复杂性。最后，本章通过实验分析了RD信誉模型对信誉评估以及资源调度带来的影响、LAGA算法的性能以及我们提出的优先级启发式的效能。

第七章总结了本文的主要工作，并对下一步的工作进行了展望。

第二章 信任管理的相关研究工作

可信作为系统能够稳定发展的基本要求，近年来在学术界和工业界都得到了广泛关注。本章首先对信任及信任管理的概念进行总结和分析，然后介绍两种基本的信任管理技术：基于身份策略和基于行为信誉的信任管理。

2.1 信任及信任管理

2.1.1 信任

为了能够更好的理解信任概念的内涵，本小节将从信任的定义、性质以及分类三个方面对信任进行解释。

1．信任的定义

信任作为人类社会的一个基本因素，它在社会组织中起到了决定性的作用。例如在一个交通十字路口时，我们总会相信另一个方向的汽车会遵守信号灯的指示，从而做出我们的决策并得以顺利出行。正是由于信任的重要性，信任研究得到了各个领域的关注，包括心理学、社会学、哲学等，随着时代的发展，又融入了商业管理、经济理论、工程学、计算机科学等应用领域知识。由于信任具有复杂性和多面性的特点，目前学术界和工业界对于信任还没有一个精确的、广泛可被接受的定义，它通常被作为一种直觉上的概念加以理解。

围绕着信任，目前存在着各种定义。牛津大辞典中信任被定义为“对某人或某物在可靠性、真实性、能力和实力方面的一种信念”。Mayer[1]将信任定义为“基于对被信者某个行为的预测，信任方愿意接受相信对方的风险，而不管能否监视或者控制被信方”。该定义着重强调了信任的风险，说明信任本质上是对接受风险的一种评估。

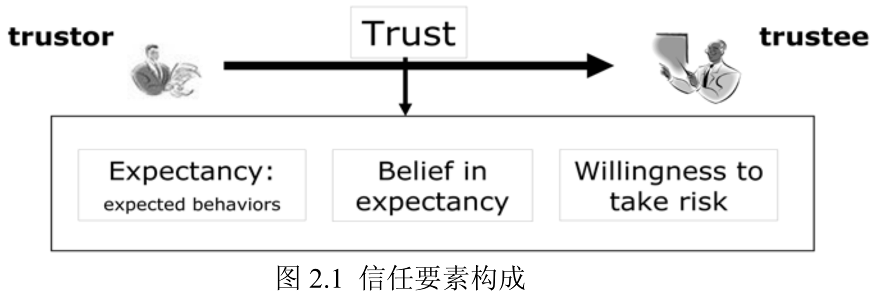

图 2.1 信任要素构成

如图 2.1 所示，Huang[2]等人给出定义“信任是一种心理状态，它包括三个方面：(1)期望：信任者希望从被信任者获得的服务；(2)信念：基于对被信任者能力

和意愿的判断，信任者相信期望是正确的；(3)风险意愿：信任者愿意承担信念可能的失败”。

D.Gambetta [3]给出如下定义：“信任（或不信任）是一方评价另一方或另一团体实际行为的主观可能性程度，评价是在对该行为进行监控（或根本不可能监控该行为）之前和与该行为对其自身行为产生影响的情况下进行”。该定义给出了信任的几个重要特征：(1) 主观性，不同的个体对同一事物的看法会受个体喜好等因素影响而不同；(2) 可能性预期，信任的程度可表示为对事件发生概率的可能性估计；(3) 上下文相关，信任是对事物的某个方面而言的。综合上面各种信任定义对信任属性的理解，我们将信任定义为：

定义 2.1 信任（Trust）：信任是在特定时间段和特定上下文环境中，授信方（Trustor）对受信方（Trustee）的某种服务属性（service utility）在诚实性（honesty）、安全性（security）、可靠性（reliablity）以及可依赖性（dependability）方面的一种主观肯定。信任也称为可信。

我们的定义限定了信任作用的时间（特定时间段）和条件（特定上下文环境），考虑了信任的本体构成（授信方，受信方及某种服务属性），明确了信任的关注因素（诚实性、安全性、可靠性以及可依赖性）。服务属性的诚实性关注受信方是否真正拥有它所宣称的服务能力；安全性关注受信方是否能安全（保密和完整等）地提供服务；可靠性关注受信方提供服务的失败率；可依赖性关注依赖受信方的风险。信任本身并不是事实或者证据，而是对于所观察到的事实的知识，而且它往往和一定的信任等级相联系[5]。信任等级是对一个实体相信程度的度量。信任等级可以是连续的，也可以是离散的，本文用信任度的概念加以描述。

定义 2.2 信任度（Trustworthiness）：信任度是授信方对受信方信任程度的量化表示，也可以称为可信度、信任值、信任级别等。

2．信任的性质

信任的动态性是信任评估和可信赖性预测的最大挑战，它是由信任关系中的实体的自然属性决定的。在现实世界中，信任的动态性(变化)既可以由实体的内因(endogenous factors，例如实体的心理、性格、知识、能力、意愿等)引起，也可以由实体的外因(exogenous factors，例如实体表现出的行为、策略、协议等)引起。信任主体的内因很难由其他主体来判断和量化(即使非常有经验的心理学家也很难做到)，而外因可以直接观察到，是可以进行预测、量化和推理的，也可以管理它们。在信任关系中，某一时刻采样到的外因特征值是一个相对静态和稳定的量，采样的时间粒度决定了推理的准确程度。外因是内因的外部表现形式，可以间接地根据外因去评估内因。因此，信任的动态性是可以量化的，信任与各种因素的关系

可用图 2.2 进行说明[6]。

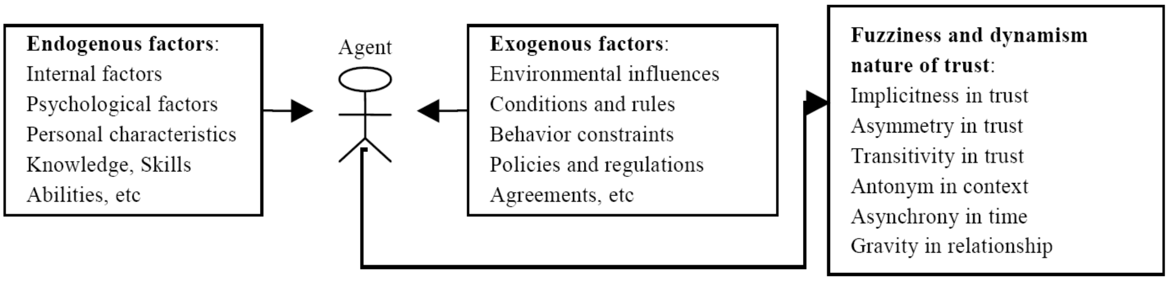

图 2.2 信任的动态性关系

总结各种文献对信任性质的认识，综合考虑影响信任动态性的各种因素，本文将信任的性质归纳如下：

（1）主观不确定性：指随着上下文和时间的变化，授信方(trustor)不能清楚地判断受信方(trustee)的动态变化，只能根据以前的交互历史对信任进行评估；信任是授信方对受信方的一种主观判断，不同的实体会具有不同的判定标准。即便对于同一受信方，相同上下文环境，相同时段以及相同行为，授信方的不同，给出的量化判断也很可能不同。

（2）上下文相关性：信任的具体状况与上下文环境紧密相关，离开具体上下文环境讨论信任问题是毫无意义的。

（3）不对称性：也就是信任关系是单向的，A 信任 B，不表示 B 也信任 A。

（4）不完全传递性：信任关系一般不具有完全传递性，即 A 信任 B，B 信任 C，不一定能得出结论 A 信任 C。只有在某些特定的约束条件下，信任才具有一定的传递性。推荐是比较典型的信任传播方式，是信任传递性的一种体现。文献[2]通过形式化的描述说明关于能力的信任不具有传递性，而基于观念的信任具有传递性。

（5）时间异步性：指实体之间的对信任关系的评估结果具有时间异步性，解决问题的办法是对时间槽进行平均； 信任会随着时间的推移而衰减，最为直接的表现是：越久远的信任评价，其说服力越差。

（6）多客面性：信任往往和受信方的多种属性相联系，受到多种属性的影响，是一个多属性作用融合的概念。以在线购物为例，顾客对卖家的评价可能包括对其产品的质量、价格、服务态度和快递的速度等多个方面的评价。

3．信任的分类

根据不同的标准，信任可以有不同的分类方法。Zucker[7]在对美国经济领域的信任演化机制进行研究后，将信任分为三类：（1）基于过程的信任，它来自于用户行为的历史记录；（2）基于特征属性的信任，它来自于用户之间特征属性的相似性；（3）基于机构的信任，它来自于用户所在机构的组成及法规等。根据信任

的内容、主体和客体的不同信任可被分为：服务提供信任、资源访问信任、代理信任、证书信任和架构信任（上下文信任）[8]。

文献[9]提出了另外一种信任划分：身份信任（Identity Trust）和行为信任（Behavior Trust），身份信任与实体身份认证的真实性有关，而行为信任与实体的声誉有关。这种划分与基于证书和策略的信任机制和基于信誉的信任机制这两种信任管理技术形成了对应关系。

根据 Donovan Artz[10]等人的观点，信任根据获取途径可分为：基于凭证的信任和基于信誉的信任。前者又称为理性信任或者客观信任，后者又称为感性信任或者主观信任。理性信任，指授信方在一定环境中相信受信方会以一定的方式执行或者不执行某项活动。它的特点是相对精确、客观，表达为信任或不信任的两种选择，信任与活动没有直接的关系。理性信任可以抽象为 0 或 1 的关系，也可以用布尔值表示。这种信任关系本身就决定了信任或者不信任的条件，信任管理则根据信任关系判断“是”或者“否”。

感性信任，则主要从信任的定义出发，使用数学的方法来描述信任意向的获取和评估。感性信任模型认为，信任是主体对客体特定行为的主观可能性预期，取决于经验并随着客体行为的结果变化而不断修正[3]。在感性信任模型中，实体之间的信任关系分为直接信任关系和推荐信任，分别用于描述主体与客体、主体与客体经验推荐者之间的信任关系。也就是说，主体对客体的经验既可以直接获得，也可以通过推荐者获得，而推荐者提供的经验同样可以通过其他推荐者获得。感性信任模型放弃了实体间的固定关系，认为信任是一种经验的体现。它对信任进行量化或者分级，通过经验、信誉或者风险分析来给出可信的概率，往往用[0,1]之间的实数或者不连续值来表示。感性信任模型所关注的内容主要有信任表述、信任证据过滤、信任度量和信誉评估。信誉评估是整个信任模型的核心，因此感性信任模型也称信誉评估模型。

理性信任和感性信任是信任管理在实际应用中的具体体现，并不存在矛盾[130]。单纯地把信任限定为某一种类型的观点是存在局限性的[48,49,50]，也不能因为某种信任否定另一种信任，两种关系决定了不同的信任方面，在信任系统中是可以并存的。而且，通过这种并存关系可以把确定因素的和不确定因素相结合，实现更加有效的信任管理。我们所需要考虑的，更多的是根据特定的信任场景，决定采用何种信任关系和何种形式的信任模型。

2.1.2 网络系统信任

关于可信，以前的大多数研究工作只是对两个实体之间的信任关系进行研究。但是随着人类社会的发展，人们需要更多地与经济社会中的陌生人交往，这就对

系统的信任提出了要求。计算机互联网络系统作为人类活动的反映，网络系统的信任正变得越来越重要。现实的发展对网络安全提出了更高的要求，希望在保障信息私密性、完整性和可用性的同时，保障网络系统的安全性、可生存性和可控性。然而，目前互联网中普遍存在的脆弱性导致了它是不可完全信任的。

尽管人们提出可信系统的概念已经有一段历史，但是国际上对可信网络系统的探索才刚刚开始，基本概念和科学问题的认识还不深入。自上世纪 70 年代初期，Anderson J. P.首次提出可信系统（Trusted System）的概念以来[11]，IT 系统的可信性问题就一直受到学术界和工业界的广泛关注。目前国际上对系统可信性比较有代表性的阐述主要有：ISO / IEC154O8[12]标准中指出，一个可信的组件、操作或过程的行为在任意操作条件下是可预测的，并能很好地抵抗应用程序软件、病毒以及一定物理干扰所造成的破坏；Microsoft、Intel、IBM 和 HP 等公司组成的可信计算组织（Trusted Computing Group）[13]认为：如果一个实体总是按照其设定目标所期望的方式行事，则称这个实体为可信的；从用户体验的角度，微软公司比尔盖茨认为可信计算是一种可以随时获得的可靠安全的计算，并包括人类信任计算机的程度，就像使用电力系统、电话那样自由、安全[14]；Algridas 和 Laprie 等人则将系统可信性表述为系统提供的服务可以被论证为可信任的，系统能够避免出现不能接受的频繁或严重的服务失效[15]。

从网络行为的角度，林闯等人认为可信的网络意味着网络系统的行为及其结果是可以预期的，能够做到行为状态可监测，行为结果可评估，异常行为可控制[16]。具体而言，网络的可信性应该包括一组属性，从用户的角度需要保障服务的安全性和可生存性，从设计的角度则需要提供网络的可控性。不同于安全性、可生存性和可控性在传统意义上分散、孤立的概念内涵，可信网络将在网络可信的目标下融合这三个基本属性，围绕网络组件间信任的维护和行为控制形成一个有机整体。

如图 2.3 所示，系统信任信息的维护过程可以分为信任信息输入、信任信息处理和信任等级或策略输出三个部分[16]。信任信息采集提供具体的输入方式，主要包括：集中式安全检测，即通过在网络中设置专门的服务器，对某个范围内的网络节点进行脆弱性检测等信任信息的采集；分布式节点自检，即将部分监测功能交由网络节点中的代理完成，网络只负责接收检测结果；第三方通告，即由于不能直接对被测节点进行检测等原因，而间接地获得有关信息。信任信息经过存储、传播和分析后，通过信任等级和策略输出用于驱动和协调需要采取的行为控制。典型的行为控制方式有：访问控制，即开放或禁止网络节点对被防护网络资源的全部或部分访问权限，从而能够对抗那些具有传播性的网络攻击；攻击预警，即向被监控对象通知其潜在的易于被攻击和破坏的脆弱性，并在网络上发布可信性

评估结果，报告正在遭受破坏的节点或服务；生存行为，即在网络设施上调度服务资源，根据系统工作状态进行服务能力的自适应调整以及故障的恢复等；免疫隔离，即根据被保护对象可信性的分析结果，提供到网络不同级别的接纳服务。

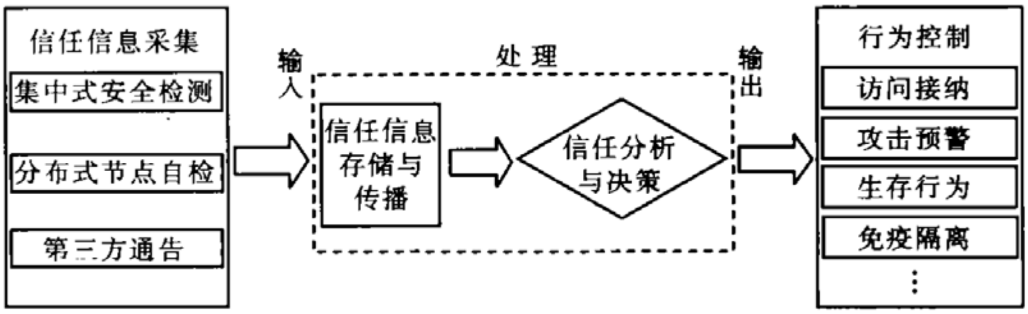

图 2.3 信任信息处理过程

文献[17]中，软件系统的可信被定义为“如果一个软件系统参与某应用环境的行为总是与预期相一致的，则称其在该应用环境中为可信（Trustworthy）的”。它们关注的焦点包括 1)身份可信问题：从安全的角度，要求资源的访问和利用总是处于规则允许的范围之内，其核心是基于身份确认的访问授权与控制；2）能力可信问题：从可依赖的角度，要求软件系统的功能是稳定和可靠的，对于客观失效（如网络故障、系统部件失效）造成的系统故障具有较好的承受能力，其核心是软件系统运行的可靠和可用性；3）行为可信问题：要求软件系统的行为总是处于系统限定的范围之内，或者系统对其的行为评价总是与对其承诺相一致。

总结上面对系统可信的理解，我们将网络系统可信定义为：

定义 2.3（网络系统的可信） 如果一个分布协同的网络系统能够确保其系统行为是负责任的（accountable）、可预期的（predictable）以及错误容忍的（endurable），则称该系统是可信（trustworthy）的。

在我们的定义中，系统的可信首先表示的是系统的行为是负责任的，即系统中各要素对自己的行为结果负责，这主要包括要素实体身份和行为记录的真实、恶意行为的自动监测和系统通告。其次，系统的可信要求系统行为具有可预测性，即系统可以提供满足用户 QoS（时间、开销、满意度等）期望的服务。另外，可信的系统还需保证一定的健壮性，即在系统的某些元素发生错误后，系统服务仍然能够保证一定质量的 QoS 服务。

2.1.3 信任管理

为了能够在分布协同的网络环境中对信任关系进行有效处理，信任管理（Trust Management）[18,19]的概念被提出。它的基本思想是承认系统中信任信息的不完整性，系统的安全决策需要依靠可信第三方提供附加的安全信息。信任管理的概念

最初由 AT&T 实验室的 M. Blaze 等人于 1996 年提出[18]，旨在“采用一种统一的方法描述和解释安全策略、安全凭证和用于直接授权关键性安全操作的信任关系”。基于该定义，信任管理的内容包括：制定安全策略、获取安全凭证、和判断安全凭证集是否满足相关的安全策略等。信任管理要回答的问题可以表述为“安全凭证集 C 是否能够证明请求 r 满足本地策略集 P”。在一个典型的 Web 服务访问授权中，服务方的安全策略形成了本地的信任权威，服务方既可使用安全策略对特定的服务请求进行直接授权，也可将这种授权委托给可信任第三方。可信任第三方则根据其具有的领域专业知识或与潜在的服务请求者之间的关系判断委托请求，并以签发安全凭证的形式返回委托请求方。最后，服务方判断收集的安全凭证是否满足本地安全策略，并做出相应的安全决策。Winsborough 等人[51]称这类信任管理系统为基于能力(capability-based)的授权系统，它们仍需要服务方预先为请求方颁发指定操作权限的信任证书，无法与陌生方建立动态的信任关系。Li 等人提出了一种基于角色的信任管理框架(role-based trust-management framework，简称 RT)[55,65]，代表了信任管理的最新研究水平。

由于信息安全的隐患源于多个方面，在利用属性信任证书和访问控制策略进行信任建立的过程中，可能泄露交互主体的敏感信息。另外在具有多个安全管理自治域的应用中，为了实现多个虚拟组织间的资源共享和协作计算，需要通过一种快速、有效的机制为数目庞大、动态分散的个体和组织间建立信任关系。而这种信任关系常常需要动态地建立、调整，并依靠协商方式达成信任关系。为了解决上述问题，Winsborough 等人[51]提出了自动信任协商(automated trust negotiation，简称 ATN)的概念，并成为当前的一个重要研究方向，它通过信任证书、访问控制策略的交互披露，资源的请求方和提供方自动地建立信任关系[27,51]。

上面介绍的基于凭证的信任管理系统本质是使用一种精确的、理性的方式来描述和处理复杂的信任关系。但在信任管理思想提出之前和之后，都有一些学者，如 D. Gambetta，A. Adul-Rahman 等人，认为信任是非理性的[3,20]，是一种经验的体现，不仅要有具体的内容，还应有程度的划分，并提出了一些基于此观点的信任模型。Povey 在 M. Blaze 定义的基础上，结合 A. Adul-Rahman 等人提出的主观信任模型思想[21]，给出了一个更具一般性的信任管理定义，即信任管理是信任意向的获取、评估和实施[22]。授权委托和安全凭证实际上是一种信任意向的具体表现。在后续的研究中[19][20]，信任管理被扩展定义为：以评估和决策制定为目的的、对 Internet 应用中信任关系的完整性、安全性或者可靠性相关的证据进行收集、编码、分析和表示的行为。证据可能包括凭证、风险评估、使用经验或者推荐信息。分析过程是根据信任需求进行信任评估或计算的过程。

从上面的分析可以发现，目前的信任管理研究主要有两方面：(1)基于策略和

信任证书的信任管理，它对应的是理性信任或者客观信任关系的管理；(2)基于信誉的信任管理，对应的是一种主观或感性信任关系管理。下面两小节将分别介绍这两种信任管理技术的相关研究工作。

2.2 基于身份策略的信任管理

基于策略的信任管理技术依赖于客观的“强安全（strong security）”机制诸如签名证书和可信证书权威，目的是为了规范用户对服务的访问。基于信任证书的访问决策通常基于具有良好语义定义的机制，该机制提供了强大的验证和分析支持。下面简要阐述基于策略和信任证书的信任管理中的基本概念。

1. 信任证书

信任证书（Credential）：经过签名的信息或策略称为信任证书（也称为凭证），可简单示例为<Key-info，Policies，Signature，Validity-time>，分别表示实体的公钥信息、策略信息、颁发者实体的签名以及有效时间。信任证书必须具有可证实性和不可伪造性。按照用途的不同，信任证书可分为身份信任证书和属性信任证书。身份信任证书主要应用于对安全级别要求高的系统，如军事系统和邮件系统。其主要代表为基于 PKI 身份认证系统的 X.509[28]。属性信任证书则主要应用于方便管理和易于操作的系统，如图书资源共享管理系统和投票系统等，其典型代表为 SPKI/SDSI[41]。

2．满足性检查算法

满足性检查算法（Compliance Checking Algorithm，CCA）是信任管理系统的统一授权决策引擎，信任管理系统的授权模型语义由 CCA 实现。如何构造 CCA 算法并对 CCA 计算复杂性与语言表达能力进行适当权衡是信任管理的核心问题。

3. 授权

授权(authorization)[30]是指分析用户提交的证书，根据证书上的属性值，为用户分配访问资源的权限。用户对资源具有什么样的操作权限，或者能够享受到什么样的服务，都体现在系统对用户的授权上。在基于身份认证的信任管理系统中，对用户的授权主要是激活用户对资源的相应控制操作。例如，在 MAC[31]系统中，用户身份直接对应着权限，系统对用户的授权则直接为其分配操作权限。在基于属性认证的信任管理系统中，对用户的授权则是激活用户在系统中的角色。例如，在 RBAC[32]系统中，系统对用户的授权，则表现在将其关联到一定的角色。

4. 委托

委托(delegation)[33,34]是一种重要的安全策略，主要思想是系统中的主动实体将权限委托给其他主动实体，以便以前者的名义执行一些工作。委托具有临时性，即角色的委托只是在某个时间段内有效。在这个时间段内，使用角色的次数可能是有限的，并且可能仅在该时间段的一些周期性时间片内可用。委托管理的一个重要问题是委托角色权限的传播问题。为便于管理，必须限制权限的传播。

5. 访问控制策略

访问控制策略(access control policy)[35,51]是用来保护资源不被合法用户非授权访问，从而规范合法用户对资源的操作。访问控制策略决定了在自动信任协商中暴露哪些证书以及这些证书暴露的先后顺序。根据描述的复杂程度，访问控制策略可分为简单策略(元策略)与复合策略。简单策略是组成复合策略的基本元素，它们的关系类似于元数据与数据的关系。

6. 信任协商模型

信任协商模型(trust negotiation model)[36,37,51]是协商双方在建立信任关系中所采取的暴露证书和访问控制策略的方式。信任协商模型的选择，决定了协商双方将采用什么样的方式来释放证书和访问控制策略信息，对敏感信息个人隐私保护具有极大的影响。

本小节下面首先介绍信任策略和信任证书的描述语言，然后介绍信任证书的分布式系统管理，最后给出几个相关的例子。

2.2.1 策略及信任证书描述语言

信任管理系统一般采用统一的信任管理语言描述策略和信任证书。委派策略是信任管理系统的核心策略，能够极大的提高分布式授权的灵活性和可伸缩性。对委派策略和其它典型访问控制策略的支持程度是衡量信任管理语言表达能力的重要标准。

访问控制策略[23,36,37]规定了访问受保护资源所需提供的信任证书集。Seamons 等人[36]通过分析四类访问控制策略语言，总结出信任协商策略语言需要满足的要求。其中一项关键特征是强调单调性，因为在分布式广域协作环境中，很难判断某实体不拥有某种信任证书。例如，某策略定义：如果你不是 ACM 的会员，就可以访问某类服务。在分布式环境中，用户只要不提供 ACM 颁发的信任证书就能访问该服务，这就违背了策略定义的本意。换言之，单调性就是保证在披露信任证书减少的条件下，不会导致最终授权集的增加。对于访问控制策略的规范和管理，Leithead 等人[25]基于本体论来研究如何在避免敏感信息泄漏的同时，简化

策略的管理工作；而 Skogsrud 等人[59]借助状态机描述了 ATN 中访问控制策略的结构，并探讨了策略的生命周期管理等问题。

RT 语言是基于角色信任管理(role-based trust-management)语言的简写。RT 定义了两类实体：主体(principal)与角色(role)。主体是指由个体程序公钥等唯一证明的实体。在信任管理系统里，RT 语言[55,66]是一类语言的集合，包括：RT_0，RT_1，RT^T 与 RT^D。RT_0 是 RT 的基础，它主要用来定义角色。一个角色定义由两部分组成：头部与主体，用谓词连接起来。根据处理类型来划分，可分为如下四种：

类型 1：$A.r \leftarrow B$，定义主体 B 是角色 A.r 的成员，即 $B \in members(A.r)$。

类型 2： $A.r \leftarrow B.r_1$，定义角色 $B.r_1$ 的成员是角色 A.r 的成员，即 $members(B.r_1) \subseteq members(A.r)$。

类型 3：$A.r \leftarrow A.r_1.r_2$，定义角色 A.r 包含所有的角色 B.r2，即 $members(A.r_1.r_2)= \cup B \in members(A.r_1)\ members(B.r_2) \subseteq members(A.r)$；其中 B 是角色 $A.r_1$ 的成员；$A.r_1.r_2$ 是一个链接角色(linked role)。

类型 4：$A.r \leftarrow B_1.r_1 \cap \ldots \cap B_k.r_k$，定义角色 A.r 的成员包含了角色集$\{B_i.r_i\}$交集的所有成员，即$(members(B_1.r_1) \cap \ldots \cap members(B_k.r_k)) \subseteq members(A.r)$。

此外，RT_0 还支持简单的委托，如 $A.r \Leftarrow B{:}C.r_2$，表示 A 将其对角色 r 的权限委托给 B，B 又将该权限委托给 $C.r_2$，即等价于 $A.r \leftarrow B.r \cap C.r_2$。$RT_1$ 在 RT_0 的基础上增加了携带参数的功能，即允许增加参数来约束角色。RT^T 语言通过提供两种角色操作(角色集成$\odot$与角色互斥$\oplus$)支持职责分离(separation of duty，简称 SoD)[5,6]。RT^D 语言主要用于处理角色激活的委托，以处理角色之间的关系。

2.2.2 分布式信任证书协商管理

分布性是信任证书的本质特点，如何使授权方在分布式环境中获得足够的信任证书是信任管理系统要解决的重要问题。为此，Winsborough 等人[51]将信任证书收集问题抽象为资源请求方和提供方之间信任证书披露序列 $Q=\{C_1，C_2，\ldots，C_n\}$的构造过程，如果协商过程中每份信任证书 C_i 都是可公开的，则称 Q 为安全披露序列。在协商双方对隐私信息的自治保护技术控制下，一个显然的问题就是如何控制这条披露序列的生成。

围绕着信任证书的协商收集问题，Winsborough 等人[51]首次提出协商策略的概念，意在控制信任关系的合理建立，并提出三项约束条件：可完成性、可结束性和高效性。据此提出了两种协商策略：一种是积极策略，要求协商方在接收到协商对端披露的信息后，披露所有可满足访问控制策略保护的信任证书；另一种是谨慎策略，协商双方在披露足量的访问控制策略后才会披露所需的信任证书。积极策略往往会披露过多与信任建立无关的信任证书；而在谨慎策略中，协商者

从信任目的出发，按照严格受控的方式，通过交换指定的访问控制策略，尽可能地减少无关信任证书的披露。这两种协商策略控制的协商交互次数与两方持有的信任证书数量呈线性关系[26,51]。

Yu 等人[26]在对上述两类策略进行研究的基础上，提出了削减协商策略，该策略属于一种改进型的回溯策略，按深度优先方式对“安全披露序列”空间进行搜索。由于削减策略是一种暴力搜索策略，虽然完备但搜索代价颇为昂贵。在最坏情况下，通信量和计算量与双方拥有的信任证书数量呈指数关系。为了降低复杂度，削减策略通过监控协商方的交互状态(当对某些信任证书的请求失败且尚未达到新信任级别的情况下，采用一种避免重发对这类信任证书的请求，或对这类信任证书请求的监控机制)，能够在有效减小安全披露序列搜索空间的同时，尽可能地保证协商的成功率。经过 Yu 等人[26]的证明，削减策略是高效且完备的，其通信复杂度为 $O(n2)$，其中 n 为双方请求的信任证书数目，计算复杂度为 $O(mn)$，n，m 分别为双方拥有的信任证书和访问控制策略数目。此外 Yu 等人[26,36,38,39]采用图论的研究方式，在披露树概念的基础上，演绎出披露树策略(disclosure tree strategy，简称 DTS)来控制披露树的生成。经证明，披露树策略可以生成封闭披露树策略族。族的概念保证协商方只要从同一披露树策略族选取协商策略，就能满足协商的互操作性，封闭性则保证了协商方可以最大限定地自由选择信息的披露方式。也就是说，如果向封闭的披露树策略族增加新的披露树策略，将不会再构成一个新的披露树策略族。

信任证书的协商收集本质上就是信任证书的分布式搜索。另外，信任证书的存储策略与信任证书搜索算法是密切相关的，不同的存储策略将直接影响搜索算法的有效性和复杂性。下面介绍文献[61]对基于 RT 语言[55,56]的信任证书的存储方式以及证书链的发现技术的总结。

1．信任证书的分布式存储

分布式环境下证书的存储方式对证书链发现影响很大。信任证书协商过程中，为达到认证的目的，需要查询证书链中的各种证书，对证书的有效性进行核实；而在查找证书时，需明确证书的存储位置，这样才能做到有的放矢，提高协商效率。

在 RT 证书中，每个角色均具有两种存储类型：发布方存储(issuer-side)与接收方存储(subject-side)，每种类型均有不同的可选值。角色存储的两种类型，分别有 3 种与两种可选状态，这样就有 6 种组合。不同的组合决定了该类角色存储类型是否良好，以及是否容易被发现与检索。

单个角色具有不同的存储状态，角色表达式(角色的交并以及角色连接等)则继

承了这些存储状态，并根据单个角色的状态进行定义。一个具有良好类型(well-typed)的角色表达式满足：

(1) 实体 A 同时具有 issuer-traces-all 与 subject-traces-all 的存储形式；

(2) 角色 A.r 具有与 r 相同的存储类型；

(3) 连接角色 A.r1.r_2：当 r_1 与 r_2 同时 issuer-traces-all 时，A.r_1.r_2 为 issuer-traces-all；当 r_1 与 r_2 同时 subject-traces-all 时，A.r_1.r_2 为 subject-traces-all；当 r_1 为 issuer-traces-all 而 r_2 为良好类型，或者 r_1 为良好类型而 r_2 为 subject-traces-all 时，A.r_1.r_2 为弱良好类型；其他情况时，A.r_1.r_2 为问题类型(ill-typed)；

(4) 角色表达式的组合 $f_1 \cap \ldots \cap f_k$：当存在一个 f_i 为 issuer-traces-all，而其他均为良好类型时，$f_1 \cap \ldots \cap f_k$ 为 issuer-traces-all；当存在一个 f_i 为 subject-traces-all，而其他均为良好类型时，$f_1 \cap \ldots \cap f_k$ 为 subject-traces-all；当所有均为弱良好类型时，$f_1 \cap \ldots \cap f_k$ 为弱良好类型；其他情况下为问题类型。

2. 分布式证书链发现

分布式证书链发现算法的前提是：存在某种机制能够代替实体 A 连接到拥有与 A 相关证书的服务器主机上。涉及到 A 的证书是指：证书由 A 发布，或使用了实体 A 角色 A.r 等。在分布式证书链发现算法中，有两个主类：ProofGraph 与 ProofNode，以及三个辅助类：BlinkingMonitor，BintersectionMonitor 与 FlinkingMonitor。ProofGraph 具有四类临时变量：

(1) nodes：维护图中所有的节点。可使用哈希图来实现角色表达式到节点的映射；

(2) edges：维护图中所有的边。支持定时存储检查与新边的增加；同时也能检索所有边的状态，包括边离开节点加入产生新边等；

(3) B-proc-queue：后向处理队列。该状态表示节点等待后向处理；

(4) F-proc-queue：前向处理队列。该状态表示节点等待前向处理。

ProofNode 具有六类临时变量：

(1) B-proc-state：后向处理状态。该状态有三种可选值：未处理等待处理与已处理。创建节点时，状态为未处理；当节点进入 B-proc-queue 队列时，状态为等待处理；当节点从图上撤出或处理时，状态为已处理；

(2) F-proc-state：前向处理状态。与 B-proc-state 状态类似，也有三种可选的值，与 B-proc-state 定义类似；

(3) B-solutions：后向查找实体的集合；

(4) F-solutions：前向查找实体的集合；

(5) B-sol-monitors：后向查询对象集合，包括后向连接监视器后向交叉监视器

等；

(6) F-sol-monitors：前向查询对象集合，与 B-sol-monitors 定义类似。

算法在具体的实现中，通过将节点与边赋予不同的状态，以达到对所有的实体与证书进行处理的目的，可充分解决证书链发现的三类问题，从而达到证书链发现与检索的目的。

2.2.3 典型示例

1．PolicyMaker/KeyNote

PolicyMaker 是目前公认的第一个基于策略的信任管理系统，Blaze 等人据此提出并阐述了信任管理的思想和方法[18]。PolicyMaker 的核心是一个授权查询引擎，称为信任管理引擎。信任管理引擎的输入可以表示为三元组(A,P,C)，其中 A 是用户希望执行的操作，P 是本地策略，C 是用户提供的凭证。查询应答可以是简单的 Yes/No 结论，也可以是进一步的授权条件。PolicyMaker 查询的语法具有如下形式：

key1，key2，…，key_n REQUESTS ActionString

查询的 ActionString 是对用户发出的请求操作的描述，key1，key2，…，key_n 是请求方的公钥序列。PolicyMaker 的策略和凭证由断言描述，断言是一种描述主体间授权委派的数据结构，具有如下形式：

Source ASSERTS AuthorityStruct WHERE Filter

其中 Source 是断言的权威源。Source 为 POLICY（关键字）时，断言称为策略；Source 为公钥时，断言称为凭证。策略存储于信任管理引擎本地，凭证由 Source 对应的主体签名，可以分布存储。AuthorityStruct 包含了被授权的主体序列，其中主体可以是公钥或门限结构。Filter 定义了请求操作必须满足的条件，可以由解释执行的程序语言描述，如 Java 等。

基于 PolicyMaker 提出的 KeyNote 语言 1999 年被 IETF 接受为 RFC2704 标准[40]。KeyNote 的策略断言和凭证断言具有简明的语法，下面是一个 KeyNote 策略的例子，该策略位于将 Library 域的所有权限委派给 Alice。

Comment: Library delegates all the rights of Library to Alice.

Authorizer: POLICY

Licensees: “DSA:5601ef88” # Alice's key

Conditions: app-domain==“Library”

其中 Authorizer 字段和 Licensees 字段类似于 PolicyMaker 中的 Source 和 AuthorityStruct。KeyNote 的 Conditions 字段对 PolicyMaker 中的 Filter 做了较大简化，以一种简单的关于操作属性的表达式语言描述。KeyNote 凭证的 Authorizer

字段是公钥，并且增加了 Signature 字段，以存放 Authorizer 对本断言的签名：

KeyNote-Version: “2”

Local-Constants: Bob="DSA:4401ff92" # Bob’s key

Carol="RSA:d1234f" # Carol’s key

Comment: Alice delegates the read action on computer articles to Bob and Carol

Authorizer: “DSA:5601ef88” # Alice’s key

Licensees: Bob || Carol

Conditions: app-domain==“Library”&&action==“read”&&cat==“Computer”

Signature: <signature of the private key of Alice>

KeyNote 查询由请求方公钥、操作属性、满足性值、策略与凭证集合四部分组成，应用程序将根据查询的满足性值进行授权决策。KeyNote 的查询评价语义[40]是 PolicyMaker 的子集，递归的定义了查询满足性值的计算原理，基本思想是寻找一条从 POLOCY 到请求方公钥的委派链。

2．SPKI/SDSI

SPKI(Simple Public Key Infrastructure)和 SDSI(Simple Distributed Security Infrastructure)最初是两个独立的研究项目，两者的初衷分别是构建不依赖于 X.509 全局命名体系的授权和认证设施，两者的互补性使之合并为 IETF 的 RFC 标准[41]，一般称为 SPKI 或 SPKI/SDSI。

SPKI 继承了 SDSI 的局部名字，局部名字由主体和标识符序列组成，SPKI 的主体表示为公钥。例如，局部名字“KeyAlice’s Bob”是指公钥 KeyAlice 定义的名字空间中的 Bob，而“KeyAlice’s Bob’s friend”表示该 Bob 定义的名字空间中的 friend。局部名字不依赖于全局命名体系，通过各局部命名空间的信任关系实现更大范围内的命名体系，具有很大的灵活性和可伸缩性。

SPKI 证书包括授权证书（authorization certificate）和名字证书（name certificate），授权证书可以表示为五元组：

(Issuer，Subject，Authority，Delegation，ValidityDates)

表示 Issuer 将 Authority 字段描述的特权委派给 Subject，Delegation 决定是否允许 Subject 将 Authority 进一步委派给其他主体，ValidityDates 是证书的有效时段。SPKI 的名字证书表示为四元组：

(Issuer，Name，Subject，ValidityDates)

名字证书表达了一种名字的蕴涵机制：Subject 代表的所有公钥都具有 Issuer 定义的名字 Name，ValidityDates 是证书的有效时段。根据名字证书定义的“名字链”，可以判定一个公钥是否具有一个局部名字，或者一个局部名字可以解析为哪些公钥。

3．REFEREE

REFEREE 是 Y.H. Chu[42]等人为解决 Web 浏览安全问题而开发的信任管理系统。虽然其设计目标比较单一，但该系统可以较完整地实现信任管理模型所列出的各要素。

REFEREE 采用了与 PolicyMaker 类似的完全可编程的方式描述安全策略和安全凭证。在 REFEREE 系统中，安全策略和安全凭证均被表达为一段程序，但程序必须采用 REFEREE 约定的格式来描述。REFEREE 的一致性证明验证过程比较复杂，整个验证过程由安全策略或安全凭证程序之间的调用完成，程序甚至能根据具体需求自主地收集、验证和调用相关的安全凭证。另外，REFEREE 能够验证非单调的安全策略和安全凭证，即能够处理一些否定安全凭证。REFEREE 灵活的一致性证明验证机制一方面使其具有较强的处理能力，另一方面也导致其实现代价较高。而允许安全策略和安全凭证程序间的自主调用则存在较大的安全隐患。另外，必须看到 REFEREE 的验证结果可能会出现未知的情况。

2.3 基于行为信誉的信任管理

基于信誉的信任管理依赖于“软安全（soft security）”方法来解决信任问题。在这种情况下，信任通常基于自身经验和网络中其它实体提供的反馈（该实体使用过提供者提供的服务）。信誉的研究最初来源于社会学、经济学和心理学等领域。在计算科学领域，由于信誉的重要性，它在很多研究方向中得到关注如：分布式系统和应用、多 agent 系统、基于 web 的社会网络等。在文献[8]中，Jøsang 将信誉定义为：关于某个人或某件事的特征或者立场的大众观点。Abdul-Rahman 和 Hailes[44]将信誉定义为“基于所掌握的主体信息或其历史行为对该主体行为的预期”；Mui 等人[45]将信誉定义为一个主体基于历史行为所产生的对其意图和行为准则的感知; Chang E.等[46]认为以上对信誉的定义没有充分考虑到时间和环境对信誉的影响，为此他们将信誉定义为“主体的信誉是对所有第三方主体对其推荐信息的综合，以反映该主体的质量特征”，其中推荐是其它主体对于被评价主体的一种信任评价。

综合以上各种对信誉的定义，我们将信誉定义为：

定义 2.4 信誉（Reputation）：节点的信誉是指在给定的时段和上下文环境下，针对节点的某种服务属性，根据系统中某些节点对该节点历史行为的反馈，对该节点未来行为的一种期望值。信誉也称声誉。

定义 2.5 信誉度（Reputation_Value）：节点的信誉度是指在给定的时段和上

下文环境下，节点信誉程度的量化表示。信誉度也称为信用度或声誉度。

我们的定义突出了信誉的时间以及上下文相关性；强调了信誉的表示对象：节点的某个服务属性；说明了信誉形成的意义：根据以往行为信息预测未来行为状况；陈述了信誉的产生机制：依据系统中某些节点的反馈，可能形成系统的全局信誉，也可能是系统的局部信誉。

信誉与行为信任相对应，它与信任并不等价。信任是一个个性化的主观信念，它取决于很多因素或证据，而信誉只是其中一种因素。在文献[43]中，作者对信任与信誉之间的关系给出了一个很好的表述：利用建立在社群基础之上的关于实体以往行为的反馈，信誉系统提供了一种通过社会控制方式创建信任的途径，从而有助于对事务的质量和可靠性进行推荐和判断。

本小节下面首先介绍分布式系统中信誉信息的存储，然后介绍如何收集信誉反馈信息，最后介绍信誉的数学模型。

2.3.1 信誉信息的存储管理

信任信息的存储管理主要考虑信任节点的身份标识管理、存储的信任信息类型、信任信息的存储方式。下面我们根据文献[131]的总结，就这三方面内容进行介绍。

1．节点身份标识

为了建立信誉，节点必须具有一定有效期的身份标识。有效期越长，信誉系统对节点的信任评价准确性更高。文献[69]对各种节点身份标识方法在匿名性、实现代价、身份易变性存等方面的差别进行了详细的探讨。下面我们介绍常见的三种节点身份标识方法：

A. 使用 IP 地址作为标识

这是一种最简单的标示节点的方法，但是这种方法有很大的局限性：对 IP 地址欺骗（IP-spoofing）行为很脆弱，另外 ISPs 经常分配给节点临时 IP 地址。P2PRep 就是采用这种身份标识方法。

B. 自分配证书的方法。

这种方法允许行为良好的节点，在连接断开后使用不同的 IP 地址建立连接的情况下，重新建立信任。该方法在提供匿名性的同时能够阻止身份假冒。缺点是：恶意节点总是可以产生新的证书，使得难以区分新加入的节点是改变标示的恶意节点还是新加入的良好行为节点；并且一个更严重的情况是一个参与者可以拥有多个标识，通过伪造交互记录来抬高其中一个标识的信誉。大多数的信誉系统都

是采用这种方法，如 EigenRep、PeerTrust 等。

C. 集中式的可信身份标识分配方法。

Douceur[70]显示如果没有一个集中式的可信实体，在大规模的非集中控制的网络中建立不同的身份标识（一个实体对应一个身份标识）是不现实的。这种方法使得分离的标识和真实身份唯一绑定，可以有效的阻止恶意节点具有多个标示和当节点恶意行为被检测出来后改变标示的行为。但该方法具有两个缺点：(1)需要很高的管理代价；(2)要求节点暴露出一定的个人信息，这对一些节点往往是不能接受的。TrustMe 采用这种方法来防止节点改变身份标识。

2．存储的信任信息类型

信任信息是基于事务的评价反馈，它可被划分为两种类型：正面肯定的评价和负面否定的评价。在信誉系统中，信任信息的存储类型有三种设计选择：

A. 只存储正面肯定的事务评价。

这种方法能够刻画节点的贡献，节点没有动机去更改身份，但未能刻画节点的恶意不合作行为和有效区分新加入的节点和恶意不合作节点。文献[71]提出的 CORC 算法采用这种方法。

B. 只存储负面否定的事务评价。

这种方法能够刻画节点的恶意不合作行为，不能有效的刻画节点的贡献，不能有效区分良好行为的节点和新加入的节点。Managing Trust[72]采用基于抱怨的方法，只存储负面的事务满意度评价。

C. 既存储正面的评价又存储负面的评价。

这种方法能够全面记录节点的行为，缺点是增加了存储开销。在现有的信誉机制中大多采用这种两者结合的方式。

3．信任信息的存储方式

信任信息的存储管理方式主要分为两种情况：集中式存储和分布式存储。下面分别对这两种存储方式进行介绍。

A. 集中式存储

许多商用系统如 eBay 和 Amazon[73]都是有代表性的集中式信誉管理系统。信任信息采用集中存放的方式，通过集中的信任管理设施实现信任信息的存储、管理和维护。其优点是较为简单、高效，并且能够为应用提供比较可靠的支持。然而，由于对集中管理设施的依赖，在自组织系统（例如 P2P 文件共享系统）这类

规模巨大，结构松散，通常没有集中控制的环境中，很难得到应用。一方面，集中管理设施的存在必然带来一定的额外管理和维护成本，相对于自组织系统通常所具有的巨大规模，这显然会成为系统本身运行的一个巨大负担。另一方面，基于对大多数 P2P 文件共享系统的观察发现，系统中随时都有大量的交互行为。这意味着如果采用集中式的信誉管理方法，集中服务器势必为大量的信信任信息存储和维护请求所累，可能成为系统的性能瓶颈，并导致中心失效的问题。

B. 分布式存储

分布式存储基于系统本身，利用节点间的相互服务提供分布式的信任信息管理。它可以进一步分为三种情况：

1) 自存储自身对其它节点的评价。

节点 i 使用节点 j 提供的服务之后，把事务满意度评价存储在节点 i 自身的数据库中。许多信誉系统如 Develop Trust、Limited Reputation 等都采用这种方式。采用这种方式，当评估者向推荐者发送请求时，推荐者把相关的信任信息传递给请求者，推荐者传递的可能是经过经过自身处理的信任信息，如传递的是一个综合信任观点，而不是具体的事务评价信息。

2) 自存储其它节点对自身的评价。

如基于 R-Chain[75]的存储方法，节点 i 使用节点 j 提供的服务之后，对节点 j 的事务满意度评价由节点 j 进行保存。当一个节点 i 需要获取节点 j 的信任信息时，直接向节点 i 进行信任信息的查询，这是一种高效的方法，因为所有关于节点 j 的信息都节点 j 本地存储，用户处于自身利益的考虑也愿意跟踪他们自身的正面信任信息。然而这个方法有两个问题：首先，如果节点能够控制它的信任信息的话，节点会夸大信任评价。对这个问题一个解决方案是随机选择一些信任信息凭证，同这些发送了凭证的节点联系来验证提供的信任信息是否有效，但该方法要求接受服务的节点保存对其它节点的评价。第二个问题是节点往往不愿意保存其它关于它的负面评价信息。

3) 由特定的监管节点存储其它节点对该节点的评价。

这种存储方法使用分布哈希表（DHT）来为系统中的每个节点分配一个信任信息监管节点来存储系统中其它节点对它的评价，使用不同的哈希函数可以实现信誉信息的备份。许多信誉系统，如 PeerTrust、EigenRep 和 Managing Trust 都采用这种方法实现信任信息的管理。通过使用哈希函数，使得监管节点不知道存储的哪个节点的信任信息，保持节点的匿名性。如果节点希望得到某个节点的信任信息，它向该节点的所有监管节点发送信任信息查询请求来，根据大多数监管节

点的存储的信任信息进行信任评估。这种方法使得节点很难通过合谋来颠覆信誉系统，因为存储哪个节点的信任信息不能进行选择，并且由多个节点负责一个节点的信任信息。所以超过一半的节点监管节点是恶意节点的可能性很低。

2.3.2 信誉反馈搜集技术

为了实现信任评价，节点需要收集被评价节点的信任信息，也就是有关被评价节点的信誉推荐（反馈）。推荐信息的创建涉及把存储的经验信息以标准的形式提交给推荐请求节点。推荐信息可以包含所有的经验信息或者一个聚合的观点。著名的信誉系统 PeerTrust、Managing Trust、FuzzyTrust[87]使用前一种方法，而 NICE、REGRET、EigenTrust 使用后一种方法。采用聚合观点的方法节约带宽，具有更好的可扩展性，但是以减少透明性为代价。

现有信誉系统的信任信息收集方式通常可以分为两类[132]，一些信誉系统假设每个实体都可以访问到所有的事务或者观点信息，换句话说，信任评价基于完整的信任信息图。这类信誉系统可被称为基于全局信誉信息的信誉系统。在基于全局信誉信息的信誉系统中，同一时刻系统中所有节点获取相同的信誉信息，即完整的信誉信息。采用这种信誉信息收集方式的信誉系统通常对系统中节点的信誉信息的存储方式有较高的要求，需要能够让所有节点安全高效获得所需要的信誉信息。一般情况下，这可通过两种方式实现：一是采用集中信誉信息存储的方法，如 eBay；另一种是采用分布的存储设施，例如 EigenTrust、PeerTrust 这两种 P2P 信誉系统，采用 DHT 如 CAN、P-Grid 来实现。

另外一些信誉系统使用局部化的信任信息查找过程。它假设每个节点具有几个邻居节点，如果节点 A 希望对节点 B 进行信任评价，那么节点 A 就会向其邻居发送信任信息查询请求，并规定查询转发的深度 TTL。收到查询请求的节点根据自身的经验数据库进行如下处理：（1）如果有关于节点 B 的信任信息，那么产生关于节点 B 的推荐信息传输给节点 A；（2）检查 TTL，如果 TTL 大于 0，那么则把请求转发给邻居节点，并且 TTL 减 1，如果等于 0 则不作处理。我们可以发现，采用局部化查找方法的信誉系统，其信任评价是基于信任信息图的子图。因此，这类信誉系统可被称为基于局部信任信息的信誉系统。基于局部信任信息的信誉系统通常对系统中节点的信任信息存储方式没有特别的要求。

通过上面的介绍，我们可以发现两种信任信息收集方法都有其优缺点。基于局部信任信息的信誉系统具有更好的可扩展性。然而，基于全局信任信息的信誉系统能够访问到完整的信任信息图，可以在网络中建立一致的全局信任信息视图，因此准确性、客观性比较高，还可以避免绝大多数攻击手段造成的危害。通信负载过大是全局计算方式面临的最大问题，这可能导致模型的可用性降低。

2.3.3 信誉模型

在介绍了信任信息的存储和收集方法后，下面我们介绍利用信任信息计算节点信誉的数学模型。

1．基于局部信任信息的信任模型

局部信誉是指节点根据局部信任信息实现的信誉评价，信息来源包括直接交互经验和其他节点提供的推荐信息。总体而言，局部信誉模型相对简单，需要的信息量较少，信誉计算的代价因此也较小。然而由于信誉信息来源较少，其信誉评价的准确性较差，并且在识别欺骗行为的能力上也存在一定的不足。典型的基于局部信任信息的信任模型有 P2PRep、Develop Trust，Limited Reputation[76]等。

P2PRep 是一个针对 Gnutella 提出的一个信誉共享协议，每个节点跟踪和共享其它节点的信誉。使用提供者信誉和资源信誉相结合的方法来减少在下载使用资源过程中潜在的风险，提出了一个分布的投票算法来管理信誉。该方法假设系统中大多数节点都是诚实推荐节点，这种假设在开放的环境中并不总是成立，在某些情况下推荐可能很少，并且大多数的推荐是不诚实的。并且，提供不诚实的恶意节点通过提交大量的不诚实推荐成为主流观点，产生不正确的信任评价。

Develop Trust 是一个基于社会网络的模型，定义了信任信息收集算法，每个节点维护一个熟人集合，和节点发生过交互的节点称之为熟人，为每个熟人维护一个熟人模型，包含熟人的服务可信度和推荐可信度，基于此节点选择一部分可信的熟人节点作为邻居节点，并且，节点可以基于上述评价自适应的更新邻居节点，通常是一定的时间间隔。Develop Trust 还定义了一个信任信息收集算法，通过邻居节点相互引荐（referral）的方法来发现见证节点（见证节点（witness）指和目标评价节点发生过直接交互的节点），进而获得见证节点的推荐，使用指数平均值信任计算方法增强信任模型的动态适应能力，有效处理节点的行为改变，并且讨论了不同的欺骗模型，提出了权重大多数技术 WMA（weightedmajority Algorithm）来应对不诚实节点的不诚实反馈。WMA 算法的思想是对不同推荐者的推荐分配不同的权重，根据权重来聚合相应的推荐，并根据交互的结果来动态的调整相应权重，但这种方法面临这样一个问题：如果节点的推荐基于少量的交互或者（并且）服务质量变化很大，那么诚实的推荐节点可能被错误的划分为不诚实节点。

Limited Reputation 是针对 P2P 文件共享提出的信誉机制，每个节点维护一定数量的具有较高信任度的朋友节点，信任信息的收集采用朋友节点之间信任信息的交换来实现，采用推荐信任度等同于服务信任的方法来进行信任信息的聚合，具有和 EgienTrust 同样的问题。

2．基于全局信任信息的信任模型

全局信誉模型依靠所有节点之间的相互推荐构造基于全局信息的信誉评价，在此基础上建立全局一致的信誉视图。eBay[77]采用集中信誉信息存储的方法，它采用最简单的信誉值计算方法：分别对正面的事务评价和负面的事务评价进行简单相加，然后正面的评价减去负面的评价作为整个的信誉评价。该方法比较原始，不能有效的刻画节点的信誉。Epinions、Amazon 采用轻微改进的算法，对所有的事务评价取平均值。

EigenTrust、PeerTrust 和 Managing Trust 采用分布存储设施进行信任信息的存储和收集。这种存储方法使用分布哈希表（DHT）来为系统中的每个节点分配一个信任信息监管节点来存储系统中其它节点对它的评价，使用不同的哈希函数可以实现信誉信息的备份。EigenTrust 是一个由 Stanford 大学针对 P2P 文件共享提出的的信誉管理系统，用来抑止非法有害的文件的传播。每个节点对应一个全局信任值，该信任值反映了网络中所有节点对该节点的评价。每次交易都会导致在全网络范围内的迭代，因此，该模型在大规模网络环境中缺乏工程上的可行性。采用预信任节点和推荐可信度等同于服务信任度方法来处理合伙欺骗的不诚实推荐行为，具有一定的局限性，不能有效处理提供良好的服务同时提供不诚实推荐的恶意节点。

PeerTrust 是一个基于信任的信任支持框架，该框架包含一个自适应的信任模型来度量和比较节点的信任度。为了计算节点的信任度，定义了三个基本的参数和两个自适应的信任因子，即从其它节点接受的反馈、节点完成的事务总数、反馈源的可信度，事务上下文因子和社群上下文因子。事务上下文因子基于大小、类别和时间戳来区分事务，社群上下文因子帮助缓解反馈激励问题，并提出了基于自适应时间窗口的动态信任计算方法来处理恶意节点的动态策略性行为改变，但提出的方法不能有效的检测和惩罚反复建立信任然后进行攻击的摇摆行为节点。PeerTrust 使用个人相似度度量的方法来计算节点的推荐可信度，处理不诚实推荐，基于反馈相似度的方法会面临公共交互节点集合很小的问题，影响信任评价的准确性。

TrustGuard[96]在 PeerTrust 的基础上进行了更深入的研究，并借鉴了控制系统中 PID 控制器思想，提出了一个可靠的动态信任计算模型，但该方法仍然未能有效的检测和惩罚反复建立信任然后进行攻击的摇摆行为节点。Managing Trust 假设网络中的节点在大多数情况下是诚实的，系统中信誉使用抱怨来表达，节点获得的抱怨越多，越不可信。ManagingTrust 使用 P-Grid 完成分布的信任信息管理。另外，信任模型依赖于节点提供的信任信息的数量和质量。而理性自私的节点由于以下原因不愿意积极提供诚实的信任信息：提供反馈会增加被评价节点的信誉，

而此节点可能会成为潜在的竞争者；节点担心提供诚实的负面反馈会遭到报复；提供诚实反馈只对其它节点有利。

相对于局部信誉模型，全局信誉能够更加全面地反映系统整体对节点行为看法，因此其准确性、客观性比较高，有利于节点不良行为的识别。从基于信誉实现激励的角度，全局信誉作为与节点绑定的唯一信誉评价，相对于局部信誉，它更有利于利用网络拓扑的不对称性和节点能力的差异提供全局一致的激励。全局信誉模型的主要问题在于，由于使用了全局的信任信息，全局信誉的计算通常会产生较高的网络计算代价。信誉全局迭代产生的消息负载是全局信誉计算面临的最大问题，例如 EigenRep 模型中所采用的全局迭代的信誉求解算法，其复杂度高达 $O(n^2)$（n 为系统的规模），这在很大程度上限制了模型的可行性。另一方面，通常情况下，全局信誉模型的求解算法收敛速度也较局部信誉模型更慢。

3．信任计算数学理论基础

下面我们根据文献[8]对信任模型的分析，介绍几种重要的信任计算数学方法。

1）简单的数学公式

许多信任模型采用简单的数学公式进行信任值的计算，如事务评价的简单相加、求平均值、加权平均方法等。

最简单的计算信誉值的方法是分别对正面的事务评价和负面的事务评价进行简单相加，然后正面的评价减去负面的评价作为整个的信誉评价。这个原理使用在 eBay 信誉论坛。优点是每个人都可以理解信誉值后的原理，缺点是比较原始，不能有效的刻画节点的信誉。一个轻微改进的算法是对所有的事务评价取平均值，这种算法在许多商业网站上使用，诸如 Epinions、Amazon。这个类别的更先进模型是对所有的事务评价进行加权平均，事务评价的权重由下列因素决定，诸如，推荐提供者的推荐可信度或服务可信度、事务评价的发生时间等。许多信任计算模型采用这种简单公式的计算方法，如 PeerTrust 及 ARTrust 等。

2）可能性估计技术

可能性估计技术主要包括两种：Bayesian 模型[113]和最大似然估计（maximum likelihood estimation）技术。

Bayesian 系统以二元事务评价作为输入（肯定评价或者否定评价），利用统计方法更新 beta 可能性密度函数（PDF）。后验（更新后）信誉值计算通过结合前验（以前的）信誉值和新的事务评价来获得。信誉值以 beta PDF 参数二元组（α，β）（α 和 β 分别表示肯定评价和否定评价的数目）的形式表示，或者以 beta PDF 可能性期望值的形式表示，可选的伴随一个偏差或者信心参数。Bayesain 系统的

优点是为信誉计算提供了一个完备的理论基础，一个缺点是人理解起来过于复杂。针对 EigenRep 信任模型计算代价和通信代价高的问题，Despotovic 和 Aberer 等人[114]提出了利用最大似然估计法计算 P2P 环境下的节点信任度的方法。为了提高估计的准确性，作者引入了节点撒谎度的概念，但没有给出撒谎度的计算方法。这种方法前提假设是每个节点提供的服务质量是一个内在的参数，不发生变化，该方法不能有效处理合伙欺骗行为。

3）Belief 模型

Belief 模型的基础是 Dempster-Shafe 证据理论，简称为 D-S 证据理论[90,91]。该理论引入信任函数来度量不确定性，并引用似然函数来处理由不知道而引起的不确定性，从而在实现不确定推理方面显示出很大的灵活性，受到人们的重视。在 Belief 模型的基础上，A. Jøsang 进一步提出了主观逻辑[118]。主观逻辑“使用观点（Opinion）来表示主观的相信，是对世界的主观信任度的逻辑运算”。在主观逻辑中，实体间的信任通过观点表达，观点可以理解为将不确定的可能性作为辅助因素加以考虑的信任度量方法，因此，主观逻辑可以看作是概率计算和二元逻辑的扩展。

4）基于模糊理论的模型

信任和信誉可以表示为语言上的模糊概念，而隶属函数可以很好的描述这种不确定度。D. W. Manchala 在文献[115]中提出了基于模糊逻辑的信任模型。该模型使用加权信任矩阵（Weighted Trust Surface）和模糊信任矩阵（Fuzzy Trust Surface）两种信任矩阵来表示两个实体在事务中的信任关系。同时，模型中包含

模糊逻辑推理，通过使用信任传播技术获得信任矩阵。可以对模糊矩阵应用模糊推理来执行不同的动作：验证、补偿等。模型还定义了 Zadeh 复合推理规则（Zadeh’s Compsitional Rule of Inference），用来解决信任矩阵的建立问题。文献[87]在 P2P 和 Grid 中提出使用模糊逻辑的信任模型和管理机制。

5）流模型

通过循环或者任意长链的可传递迭代来计算信任或者信誉，这类模型称之为流模型[8]。

一些流模型假设整个社群具有一个常量的信任或者信誉权重，这个权重分布在社群中的所有成员。参与者的信誉值的增加以其它节点信誉的降低为代价。Google 的 PageRank[116]和 Advogato[117]的信誉算法属于这一类别。通常，一个参与者的信誉增加作为一个输入流函数，减少作为一个输出流函数。在 Google 中，指向一个网页许多超链接会为 PageRank 的增加作出贡献，而从一个网页发出的许多

超链接会减少该网页的 PageRank。流模型并不都要求信誉或者信任值的和等于一个常量，这样的例子如 EigenTrust 模型，计算 P2P 网络中节点的信任值沿着传递链重复和迭代相乘和聚合获得，直到所有节点的信任值聚合到稳定值。

2.4 小结

本章对信任的相关概念和工作进行了介绍。在分析已有信任定义的基础上，本章给出了我们的信任定义，并分析了信任的基本性质以及它的各种分类方法。介绍了网络系统信任的定义和内涵，分析了信任管理的需求及其种类。针对基于策略和信任证书的信任管理，本章介绍了它的相关概念，以及策略信任证书的描述语言和分布式管理方法，并介绍了三种典型的基于策略的信任管理系统。针对基于信誉的信任管理，本章给出了相关定义，介绍了信任信息的存储和收集方法，以及信誉的多种数学模型。

第三章 面向身份策略的信任描述语言及证明算法

随着分布式网络应用的日益普及，如何对分布式环境中跨安全域的资源进行访问授权管理，成为当前的研究热点。其中一个主要的研究方向为自动信任协商(automated trust negotiation，简称 ATN)。与传统的基于访问控制列表的方法相比，它的特点是通过信任证书和访问控制策略的交互披露，资源的请求方和提供方自动地建立信任关系[51]。目前，对 ATN 的研究主要有两方面：信任证书的分布式放置收集方法以及策略语言的形式化描述分析。ATN 策略语言的内容可分为资源访问控制策略和信任协商策略，一般情况下，策略语言将两者分离并分别进行描述[52]。

信任证书分布式放置收集方法包括传统的信任协商和分布式证明两种方法[53]。Bauer[54]分析认为，分布式证明方法比传统信任协商方法更高效，并能克服后者的一些缺点如信任求证节点负载过大；策略语言的资源访问控制策略利用角色对权限的授权和委托信息进行定义，角色的描述能力越强对应用的支持就越大；策略语言的信任协商策略需提供对信任证书敏感信息的保护并避免信任证书的盲目搜索。综合上面三方面的特点，开发出一种策略语言具有强大的资源访问控制描述能力，能够保护信任证书敏感信息，避免信任证书的盲目搜索并能支持信任分布式证明成为本章的目标。

目前一些研究工作都只是对我们目标的某个方面提出解决办法，如文献[54]通过定义 say 谓词提出了一个信任分布式证明算法，但该算法不支持复杂的资源访问控制和信任证书信息保护；Li 在文献[55]中提出一种资源访问控制策略 RT(role-based trust-management)语言，支持参数化和连接两种复杂角色，但不能支持敏感信息保护；Li[56]在 RT 语言的基础上通过加密信任证书技术支持敏感信息保护，但这两者都不能支持信任的分布式证明方法。以上语言和方法之所以不能对信任分布式证明方法以及复杂的资源访问和信任协商策略提供全面支持，主要是缺少一个功能全面的策略描述语言。

本章提出一种面向信任分布式证明和协商的策略语言 RTP(Role-based Trust Proving)。RTP 语言的主要特点如下：1）对 RT 语言进行改进和功能拓展，从而使 RTP 语言的访问控制策略能够延续 RT 语言描述能力强的优点，可以定义复杂的角色如连接角色和带参数的角色。2）语言中增加 lsign 语法，可以定义逻辑推导角色，且该新增类型角色是对传统角色定义的放松，能够支持信任分布式证明。3）语言的信任协商策略增加 release 谓词，从而可以限制信任证书信息的传播，提供对信任证书保护技术的支持。4）信任协商策略中增加 prove 和 find 谓词，可以定

义信任协商启发式规则，从而避免信任证书的盲目搜索。文章详细介绍了 RTP 语言的语法构成，定义了 RTP 语言的推理证明规则，给出了语言的语义解释并证明了语言的可靠性和完全性。

为了体现 RTP 语言的全面功能，本章还提出一个基于 RTP 语言的信任分布式证明协商算法 DPN(distributed proving and negotiation)。DPN 算法通过本地信任协商和远程信任证明，可以高效地完成信任分布式证明任务。另外通过信任证书限制策略和启发式规则，算法可以有效地保护信任证书敏感信息，并避免信任证书盲目搜索。文章通过信任图的概念分析了算法的正确性和完整性。我们的实验表明，跟传统的信任协商方法相比，DPN 算法能够有效地减少信任建立时间和交互次数。

本章第 1 节介绍研究相关工作。第 2 节给出 RTP 语言的框架和语法。第 3 节介绍 RTP 语言的推演规则和语义。第 4 节描述分析一个 RTP 语言信任分布式证明协商算法。最后一节总结本章工作。

3.1 相关工作

资源访问控制策略作为策略语言的基本要素，主要描述资源访问权限的授权和委托信息，文献[55~60]从不同方面(如义务[57]及角色管理[58])对该策略进行描述。信任协商策略是协商双方在建立信任关系中所采取的暴露访问控制策略的方式[61]，包括协商过程中如何决定向谁索取信任证书、索取什么信任证书；另外一个重要方面是对信任证书敏感信息提供保护[62]，如我们不想让别人从身份信任证书中知道自己的年龄，涉及的技术主要有隐藏信任证书[63]和无记忆属性证书[64]等。

文献[55,65,66]系统地介绍了 RT 系列语言，语言中角色由实体和角色名组成，如 Bob.Fridend 表示实体 Bob 定义的角色名为 Fridend 的角色。RT 语言引入了 4 种基本访问控制策略，定义了参数化角色如 Bob 的男性朋友(Bob.Fridend(boy))以及连接角色如 Bob 朋友的老师(Bob.Friend.Teacher)等描述功能强大的角色，能够很好地满足应用需求。该语言的缺点是不能对信任证书的敏感信息提供保护。J.Li[56] 在 RT 语言的基础上通过信任证书加密技术对访问策略的敏感信息提供保护，但这两者都没有解决协商中盲目索取信任证书的问题。

文献[51]定义了传统的分布式协商模型，信任请求方不断向信任证书拥有者请求下载信任证书，信任证书拥有者判断请求是否满足策略要求，不满足则向请求方发送要求，请求方相应做出反应。如此不断迭代，最终建立信任关系。例如一个信任授权中心 CAS 有 RT 语言描述的规则 CAS.trust←CAS.honor 和 CAS.honor←Alice，意思是 CAS 将 Alice 认证为自己尊敬的成员，尊敬的成员也是

信任的成员。当 Alice 向 CAS 求证自己是其信任成员时，两者之间的传统协商过程如下：1) Alice 发送认证请求(?CAS.trust←Alice)；2) CAS 返回要求(?CAS.honor←Alice)；3) Alice 发送请求(?CAS.honor←Alice)；4) CAS 返回信任证书(CAS.honor←Alice)；5) Alice 发送信任证书(CAS.honor ←Alice)；6) CAS 认证成功并发送信任证书(CAS.trust←Alice)。如图 3.1 所示，协商的前半部分(步骤 1~步骤 3)都是交换请求，后半部分(步骤 4~步骤 6)则交换信任证书。基于 RT 语言的方法[5,6,65,66]也采用传统的分布式协商方法，它基于信任证书类别收集信任证书，收集策略包括前向搜索，后向搜索和混合搜索。

传统的信任协商存在如下问题：1) 授权服务器必须时刻处于工作状态，对信任签名请求进行判断并给出签名。这可能导致拒绝服务攻击，不符合信任管理给服务器减负的思想；2) 信任请求方向其他节点索取信任证书时，很可能因为索取太多权限或太少权限造成失败[52]；3) 信任请求方进行信任的全部证明工作，且当有多条访问策略匹配时，只能盲目索取相关的所有信任证书，使信任请求方成为影响系统性能的瓶颈。

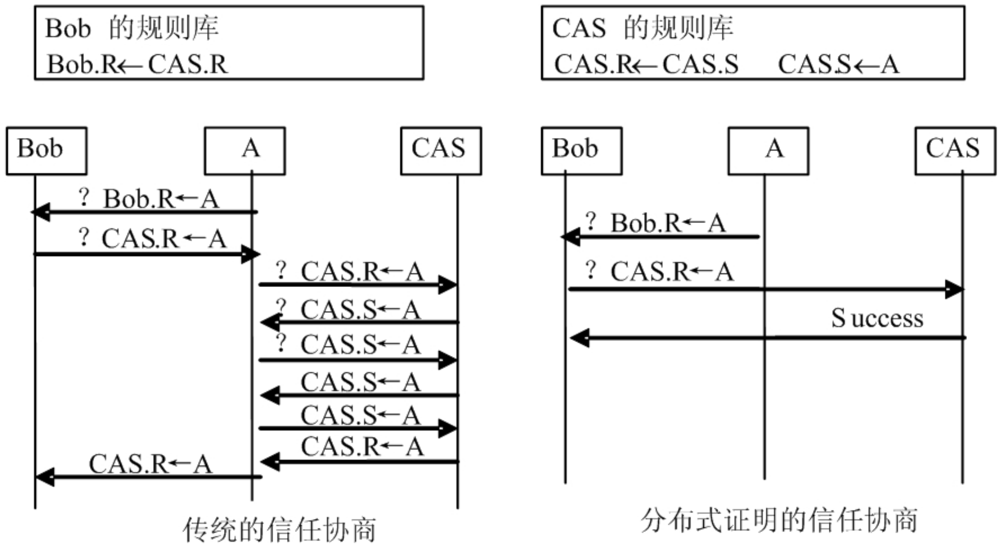

图 3.1 信任协商示例

上文的例子中，CAS 很显然会得到 CAS.trust←Alice 的结论，但由于传统信任协商中服务方只提供信任证书签名服务并不为请求进行证明，因此 Alice 和 CAS 进行了很多无谓的交互工作。图 3.1 右表示的是当信任协商中支持分布式证明时的协商场景。信任服务方为请求进行力所能及的推理证明，Bob 为请求(?Bob.R←A)直接向 CAS 求证(?CAS.R←A)，CAS 用自己的规则库对请求进行证明并返回成功的消息。显然支持分布式证明的信任协商更高效。

文献[52,54,67]讨论了信任的分布式证明方法，思想是信任服务方为请求进行力所能及的逻辑证明，从而减少信任签名请求并减轻信任请求方的信任协商负担。

PeerAccess[52]给出一个了支持信任分布式证明的语言，但它不能支持 RT 中的参数化角色和连接角色，没有给出分布式信任证明的实现方法。Lujo[54]通过定义 say 语法支持信任分布式证明，并给出一个分布式证明算法，实验从远程请求次数和信任证书拥有者被访问次数两个角度表明了分布式证明带来的性能提升。它的缺点是系统策略语言只能实现简单的资源访问控制和委托，不提供信任证书敏感信息的保护，算法没有给出证明过程中证明规则。

3.2 RTP 语言框架及语法

本节在 RT 语言的基础上进行拓展形成 RTP 语言，它的新特点是能够支持信任的分布式证明、信任证书信息保护以及证明的启发式规则。信任证书可以认为是策略规则的数字签名[68]，因此对信任证书和策略规则可采用统一的语言进行描述。本节首先介绍 RTP 语言所需描述知识的框架，然后介绍适应该需求的 RTP 语言语法。

3.2.1 RTP 语言知识框架

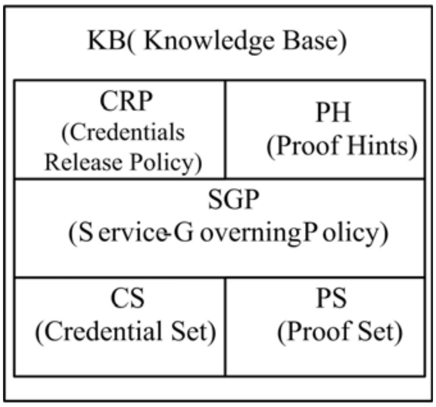

图 3.2 RTP 语言局部知识库框架

在信任协商中，节点(principal)表示一个进程或 Agent。它也可以由一个公钥表示，能够对信任信息进行签名或发出信任请求。系统中所有的节点构成一个集合 N。每个节点 A 都有一个局部知识库 KB(Knowledge Base)记为 P_A，系统中所有节点的局部知识库的集合组成系统的全局知识库 P。我们的目标信任协商系统需要节点的局部知识库 KB 支持资源访问控制，信任证书信息保护，信任协商证明启发式以及信任的分布式证明功能。因此我们将 KB 设计成一个通用可扩展的结构，如图 3.2 所示，它包括五个部分：信任证书释放规则 CRP，证明启发规则 PH，服务访问控制规则 SGP，信任证书集 CS 和证明结果集 PS。其中 CRP 和 PH 构成 KB 的信任协商策略，SGP 构成 KB 的访问控制策略。各部分介绍如下：

- 信任证书释放规则定义信任证书的输出条件，保护信任证书敏感信息。
- 证明启发规则指出信任协商时的推理规则，如向谁索取信任证书，索取什么信任证书。
- 服务访问控制规则是本节点定义的基于角色的访问控制规则。
- 信任证书集包含协商过程中从其它节点收到的直接签名信任证书。
- 证明结果集包含自己或从其它节点收到的证明中得到的逻辑结论。

为了便于理解局部知识库和 RTP 语言，图 3.3 例 1 给出一个书店 Org 的局部知识库，这里我们只介绍各规则的直观含义，语法由下一小节介绍。CS 缓存的直接签名规则 cs1 表示 Alice 是书店管理员 Reg 的贵宾。PS 的缓存结论 ps1 表示 Alice 相信 Bob 是自己的信任者。服务访问规则 s1 规定被管理员认为积分大于 1000 的顾客是书店 Org 的贵宾；s2 规定管理员认为的贵宾的信任者是该书店的信任者；s3 规定 Carla 是该书店的一个贵宾。信任证书释放规则 c1 表示，只有书店 Org 的贵宾能把 s2 规则发送给其它节点，且只能传给书店的信任者。证明启发式规则 p1 表示根据信任证书 cs1 和规则 s2，要证明某人是 Org 的信任者，还需证明该人是 Alice 的信任者。p2 表示要证明某人是 Alice 的信任者，须向 CAS 求证。

机构的 CS
cs1. Reg(sign). honored ←Alice

机构的 PS
ps1. Alice(lsign).trust ← Bob

机构的 SGP
s1. Org(lsign).honored ← Reg(lsign).customer(score>1000)
s2. Org(lsign).trust ←Reg(lsign).honored.trust
s3. Org(sign).honored ← Carla

机构的 CRP
c1.Org(lsign).release(s2, B, A)←(Org(lsign).honored←B ∩ Org(lsign).trust←A)

机构的 PH
p1. Org(lsign).prove(Org(lsign).trust , {cs1, s2} , {Alice(lsign).trust})
p2. Org(lsign).find(Alice(sign).trust , CAS)

图 3.3 例 1 Org 的局部知识库 KB

3.2.2 RTP 语言语法

图 3.4 给出了 RTP 语言的 BNF 描述，下面解释中出现的数字指的是图 3.4 中该数字行对应的语法。系统中每个节点都有一个局部知识库 Knowledge-base(4)，它由三种类型的规则组成：服务访问规则，信任证书释放规则和证明启发式规则。

1. 服务访问规则 SGP

服务访问规则(5)由头部和尾部组成，规则头部为一个角色 f，规则尾部(6)有两种类型，一种只有一个节点标示 prin，表示该节点属于规则头部的角色；另一种形式为 $f_1 \wedge \cdots \wedge f_m$ ，由一组角色 f_i 的合取组成，表示如果某节点属于规则尾部所有的角色，则该节点属于规则头部的角色。第二种形式的服务访问规则省略了一个变量，等同于 $(f \leftarrow X) \leftarrow (f_1 \leftarrow X) \wedge \cdots \wedge (f_m \leftarrow X)$。

```
<set of X>::= э | <X> <set of X>                                                    1
<list of X >::= < X > | < X > “,” <list of X >                                       2
<conj of X>::= < X > | < X > “∩” < conj of X >                                       3
<Knowledge-base>::= <set of policy-expr> | <set of cred-rele> | < set of proof-hint>  4

<policy-expr>::= <role> “←” <policy-body>                                            5
<policy-body>::= <prin> | <conj of role>                                             6

<role>::= <prin> ( “(sign)” | “(lsign)” ) “.” <role-terms>                           7
<role-terms>::= <role-term> | <role-term> “.” <role-terms>                           8
<role-term>::= <role-name> | <role-name> “(” <list of field> “)”                     9
<field>::= <field-name> <relation> <constant>                                        10

<cred-rele>::= <rele-stmt>“←”“(”<conj of policy-expr>“)”                             11
<rele-stmt>::= <prin> ( “(sign)” | “(lsign)” ) “.” <release-term>                    12
<rele-term>::= “release” “(” (<policy-expr>|<cred-rele>)<prin> <prin> “)”            13

<proof-hint>::= <prin> ( “(sign)” | “(lsign)” ) “.” (<find-term>|<prove-term>)       14
<find-term>::= “find” “(”<role> “,” <prin> “)”                                       15
<prove-term>::= “prove” “(”(<role>|<rele-stmt>)“,”<grp of policy-expr>“,”<grp of role>“)”  16
<grp of X>::= “{”<list of X >“}”                                                     17
```

图 3.4 RTP 语法的 BNF 描述

角色(7)表示一个节点集合，它有两种类型：直接签名角色和逻辑推导角色。直接签名角色形式为 A(sign).R，逻辑推导角色形式为 A(lsign).R，其中 A 为节点，R 为一个角色项或角色项的连接(8)。角色项的连接由角色项之间通过点符号连接而成，表示一个连接角色，如例 1 中的 s2 规则。角色项(9)由角色名或由角色名跟一组属性域组成，属性域(10)包括域名，域值以及变量关系，域值是属性定义域中的常量。带属性域的角色也称参数化角色如例 1 中 s1 规则，Reg 是节点，customer 是角色名，score 是属性域名，1000 是域值，“>”是变量关系。下文中一般用 $R, R_1 \cdots$ 表示角色项变量，$f, f_1 \cdots$ 表示角色变量。

如果节点 B 属于直接签名角色 A(sign).R，表示节点 A 签名认定 B 属于自己的角色项 R。如果节点 B 属于逻辑推导角色 A(lsign).R，表示某节点通过知识库推导认为 B 属于 A 定义的角色项 R。传统信任协商中的角色都是直接签名角色，角色

关系必须由角色定义节点签名才有效，即信任关系须由信任定义节点亲自审核，这给信任定义节点带来负担。逻辑推导角色关系就不必要角色定义节点亲自签名，任何节点只要有足够证据就可认为信任关系成立，这是对直接签名角色的一种放松。例 1 中，由规则 s2 和信任证书 cs1 以及结论 ps1，Org 可以推导出 Org(lsign).trust←Bob 的结论，如果将 s2 规则的尾部角色改成直接签名角色即 Org(lsign).trust←Reg(sign).honored.trust，则上述结论不能成立，因为 Reg(sign).honored.trust←Bob 信任关系需由管理员 Reg 亲自审核签名才能成立，不能由其它节点推导得到。

2. 信任证书释放规则 CRP

很多情况下，信任关系的定义节点希望能控制信任证书的传播并保护信任证书的敏感信息。为此系统对 SGP 规则，CRP 规则和 CS 中的信任证书的输出进行限制，它们可以由节点 A 发送出去必须满足两个条件： i)信任证书所指的规则在节点 A 是正确的，ii)必须满足节点 A 保存的该信任证书的释放规则。

信任证书释放规则(11)由释放描述和释放条件两部分组成，释放条件是一组服务访问规则的合取，表示只有当所有服务访问规则都满足时，释放描述中的情况才准许发生。释放描述(12)的定义与角色类似，A(sign).rele_term 和 A(lsign).rele_term 分别表示直接签名和逻辑推导的访问规则，rele_term 是释放规则项。释放规则项(13)由 release 算子加参数表示，参数包括一条规则 ψ 和两个节点 A、B，表示节点 A 可以将规则 ψ 传给 B。示例中 c1 规则为信任证书访问规则的一个例子，该规则可避免 s2 规则被不相关节点得到。

一个节点只能给自己知识库中的 SGP 规则定义释放规则，当未定义它的访问释放规则时，系统默认为释放条件为真即可以释放。系统不准许节点给知识库中的 CRP 规则以及 CS 中的信任证书定义释放规则。CRP 规则的释放规则由系统按照自释放推演规则自动得到，这将在下一节 RTP 语言推演规则中讨论。CS 中信任证书的释放规则只能从它的定义节点得到，当未定义它的访问释放规则时，系统默认为不可以释放，这样信任证书签名节点定义的信任证书释放规则就能在整个系统中控制信任证书的传输。

3. 证明启发式规则 PH

证明启发式规则(14)也有直接签名和逻辑推导两种形式，包括寻找启发和证明启发两种类型。寻找启发项(15)由算子 find 和参数组成，参数包括角色 R 和节点 A，表示如果想证明某节点属于角色 R 需向节点 A 寻求帮助；证明启发项(16)由算子 prove 和参数组成，参数包括角色 f 或只含一个变量的释放规则项 RE，规则集 PS 和角色集 RS，表示如果要证明某节点属于角色 f 或规则项 RE 定义变量的值域，

根据规则集 PS，还需证明节点属于角色集 RS 中的所有角色。两种启发式规则示例见例 2 中的规则 p1 和 p2。

3.3 RTP 推演规则语义

3.3.1 RTP 语言推演规则

RTP 语言也是基于 Datalog ^{C} [66]的逻辑描述语言，因此一阶逻辑的公理定理 RTP 语言都适用。RTP 语言知识库的推演规则由定义 3.1 给出。

定义 3. 1(RTP 语言推演规则) 推演规则包括以下六条：

1) 实例推演 VI(Variable Instantiation)。如果规则$\phi \in P_A$，P_A 为节点 A 的知识库，x 为ϕ 中一个变量，b 为该变量定义域上一个常量，则$P_A \mapsto \mathrm{sub}_{\mathrm{b}}^{x}\phi$，$\mathrm{sub}_{\mathrm{b}}^{x}\phi$表示用 b 替代$\phi$中所有 x 的出现。若$\phi =$"$f \leftarrow f_1 \wedge \cdots \wedge f_m$"，$f, f_1, \cdots f_m$ 为角色，则 $\mathrm{sub}_{\mathrm{b}}^{x}\phi=$"$(f \leftarrow \mathrm{b}) \leftarrow (f_1 \leftarrow \mathrm{b}) \wedge \cdots \wedge (f_m \leftarrow \mathrm{b})$"。

2) 角色连接 RL(Role Link)。如果规则ϕ =“A(lsign).R_1←B”且$\phi \in P_A$，则 $P_A \mapsto \phi'$，ϕ' =“A(lsign).R_1.R_2←B(lsign).R_2”。

3) 假言推理 MP(Modus ponens)。如果规则ϕ =“$S \leftarrow S_1 \wedge \cdots \wedge S_m$”$\in P_A$，$S, S_1 \cdots S_m$ 为访问控制规则且$\{S_1 \cdots S_m\} \subset P_A$，则 $P_A \mapsto S$。

4) 签名推演 SD(Signature derivation)。如果一个直接签名规则$\phi \in P_A$，则 $P_A \mapsto \phi'$，ϕ' 为ϕ 对应的逻辑签名规则；如果一个被节点 A 逻辑签名的规则 $\phi' \in P_A$，则 $P_A \mapsto \phi$，ϕ 为ϕ' 对应的直接签名规则。

5) 自释放规则 1 SR1(Self_Release1)。如果规则$\phi \in P_A$，ϕ =“B(lsign).release(ψ, C, D)←$S_1 \wedge \cdots \wedge S_m$”，则 $P_A \mapsto \varphi$，φ =“B(lsign).release(ϕ, A, C)”。

6) 自释放规则 2 SR2(Self_Release2)。如果$\{\sigma, \psi\} \subset P_A$，$\sigma$=“$B$(lsign).release($\phi$, C, D)←$S_1 \wedge \cdots \wedge S_m$”，$\psi$=“$B$(lsign).release($\phi$, D, E)←$S_1' \wedge \cdots \wedge S_n'$”，则 $P_A \mapsto \varphi$，φ =“B(lsign).release(ψ, A, C)”。

上面六条推演规则中，实例推演和假言推理是一阶逻辑的通用规则，下面对其它四条规则的直观意义给出解释。角色连接规则表示如果节点 A 接受 B 节点作为自己角色 R_1 的成员，则 B 定义的角色 R_2 也将得到 A 的承认。签名推演规则可

以知道，如果一条直接签名规则ϕ在某个节点A为真，则ϕ对应的逻辑推导规则ϕ'在A也为真；如果一条A节点逻辑推导的规则ϕ'在节点A为真，则ϕ'对应的直接签名规则ϕ在A也为真。也就是说，直接签名规则可直接推出逻辑推导规则，但逻辑推导规则只有在签名节点才能得到对应的直接签名规则。

CRP 规则的释放规则只能由自释放推演规则得到。自释放规则 SR1 的意思是如果节点B授权节点C发布规则ψ，则B允许任何节点A通知C它得到的B的授权。自释放规则 SR2 在 SR1 的基础上进一步加长了信任证书的授权发布链，意思是如果节点B授权节点C发布规则ϕ给节点D，同时授权D发布规则ϕ给节点E，则B允许任何节点A将D有权给E发布规则ϕ的消息通知C，以便C将该消息转发给D。这样D有权给E发布规则ϕ的消息就可以沿着A-C-D传输，进而让D得到B的授权。在定义了上面六条推导规则后，全局知识库的推导证明可由定义 3.2 给出。

定义 3.2(推导证明)。 全局知识库P在A节点推导出规则ϕ当且仅当存在一个推导序列$P^0, P^1 \cdots P^n$，有$P^0 = P, \phi \in P_A^n$，其中$P^{i+1}(0 \le i < n)$由P^i实行一个推演规则得到。此时$P^0, P^1 \cdots P^n$称为ϕ在A节点的证明序列，表述为$P_A \mapsto \phi$。

3.3.2 RTP 语言语义

定义 3.3(知识库的 H 域解释 *W*)。 知识库P的 Herbrand 域(H 域)解释W为知识库原子集(作用在 H 域上的所有角色关系)中各原子命题真假值的设定。

知识库的 H 域包括系统中所有需要判定的对象，如系统中的所有节点。原子命题是不带变量的角色关系，因此知识库的 H 域解释W为知识库中所有角色关系作用在系统中任何节点的真假值设定。利用知识库的 H 域解释可定义知识库的解释和模型如下：

定义 3.4(知识库解释 *I*) 全局知识库P的解释I是一个包含各节点局部解释的集合$I = \{I_A \mid A \in N\}$，其中I_A是全局知识库P在A节点的H域解释集合。

定义 3.5(模型 |=) 设I为知识库P的一个解释，I_A是P在节点A的 H 域解释集合。

1）如果规则ϕ=“A(lsign).Role$_1$.Role$_2$ $\leftarrow$B”是没有变量的规则，$I \models_A \phi$当且仅当对I_A中的每个 H 域解释 W，$\exists C\,(C \in N)$: S_1=“A(lsign).Role$_1$ $\leftarrow C$”$\in W$ 且 S_2=“C(lsign).Role$_2$ $\leftarrow$B”$\in W$。

2）如果规则 ϕ =“ $S \leftarrow S_1 \wedge \cdots \wedge S_m$ ”是一个没有变量的逻辑签名规则，$I \models_{\mathrm{A}} \phi$ 当且仅当对 I_{A} 中的每个解释 W，要么 $S \in W$，要么 $\exists i\ (1 \le i \le m), S_i \notin W$ 。

3）如果 ϕ 是一个带有变量的逻辑签名规则，$I \models_{\mathrm{A}} \phi$ 当且仅当对 ϕ 进行实例化得到的每个没有变量的逻辑签名规则 ϕ'，$I \models_{\mathrm{A}} \phi'$。

4）如果 ϕ 是一个直接签名规则，$I \models_{\mathrm{A}} \phi$ 当且仅当 ϕ 是由 A 签名的，且 $I \models_{\mathrm{A}} \phi'$，$\phi'$ 是 ϕ 对应的逻辑签名规则。

5）如果 Φ 是一个规则集，$I \models_A \Phi$ 当且仅当对任意 $\phi \in \Phi$，有 $I \models_A \phi$。此时称解释 I 在 A 节点是 Φ 的模型。

对全局知识库 P，$I \models P$ 当且仅当对任意节点 $A \in N$，有 $I \models_A P_A$ 。这种情况下，我们称解释 I 是 P 的模型，记为 $\underline{P}$。模型 $\underline{P}$ 反映的是节点的局部视图，直观意义就是系统中各节点根据自己知识库可以知道的所有角色属性关系。模型 $\underline{P}$ 有下面两个定理：

定理 3.1(模型的可靠性)。对任意全局知识库 P，节点 A 和规则 ϕ，如果 $P_A \mapsto \phi$ 则 $\underline{P} \models_{\mathrm{A}} \phi$。

证明： $P_A \mapsto \phi$，设有证明序列 $P^0, P^1 \cdots P^{\mathrm{n}}$，$P^0 = P, \phi \in P_A^n$。下面用关于 $\mathrm{i}(0<\mathrm{i}\le\mathrm{n})$ 的第二归纳法证明 $\underline{P} \models_A P^i (0 < \mathrm{i} \le \mathrm{n})$，即证明 P^{i} 比 $P^{\mathrm{i-1}}$ 新增的规则在解释 $\underline{P}$ 中为真。假设 $P^{\mathrm{i-1}}$ 到 P^{i} 运行推导规则分别如下。

1) 实例推演。如果一个逻辑签名规则 ϕ 对解释 $\underline{P}$ 在 A 节点为真，则规则 ϕ 的所有实例都会出现在 $\underline{P}_A$ 的每个 H 域解释中。因此实例推演得到的规则对解释 $\underline{P}$ 在 A 节点为真。

2) 角色连接。如果规则 ψ =“A(lsign).$R_1 \leftarrow B$”和规则 ϕ =“B(lsign).$R_2 \leftarrow C$”对解释 $\underline{P}$ 在 A 节点都为真，即 ψ 和 ϕ 的任意实例化 ψ' 和 ϕ' 出现在 $\underline{P}_A$ 中的每个 H 域解释 W 中，由模型的定义可知 φ =“A(lsign).R_1.$R_2 \leftarrow C$”也将出现在 $\underline{P}_A$ 的每个 H 域解释中。因此规则 σ =“A(lsign).R_1.$R_2 \leftarrow B$(lsign).R_2”为真。

3) 假言推理。假设规则 ϕ =“ $S \leftarrow S_1 \wedge \cdots \wedge S_m$ ”对解释 $\underline{P}$ 在 A 节点为真，S' 和 $S_1' \cdots S_m'$ 为规则 ϕ 任意实例化后 S 和 $S_1 \ldots S_{\mathrm{m}}$ 对应的结果，且 $S_1' \cdots S_m'$ 出现在 $\underline{P}_A$ 中的每个 H 域解释 W 中。若 S' 不在 W 中，则得到规则 ϕ 为假，与初始假定矛盾。因此 S' 也必在 W 中，即 S 对解释 $\underline{P}$ 在 A 节点为真。

4) 签名推演规则和自释放规则本身即为语义限制所定义，因此它们推导的结果对解释 $\underline{P}$ 为真。

定理 3. 2(模型的完全性)。对任意知识库 P，节点 A 和规则 ϕ，如果 $\underline{P}\ \models_{\mathrm{A}} \phi$ 则 $P_A \mapsto \phi$ 。

证明： 系统的推演规则中，签名规则，自释放规则和角色连接规则的语义直接跟推导证明的语法对应，因此 $\mapsto$ 语法可直接得出。实例推演和假言推理的证明与一阶逻辑证明一样，可以通过一阶逻辑的协调性来证明，篇幅有限，这里不做具体展开。

3.4 基于 RTP 语言的信任分布式证明算法

为体现 RTP 语言的特点，本节介绍一个利用 RTP 语言及其推演规则实现分布式证明协商 DPN(Distributed Proving and Negotiation)的算法，并分析该算法的性能。

3.4.1 DPN 算法描述

DPN 算法如图 3.5 所示，recursive_prove 函数递归的将目标分为子目标进行证明。算法输入为一个证明目标列表，目标的表示形式与 SGP 规则一样包含头部 h 和尾部 b。输出为证明目标所需的知识库中的规则，如果协商失败，输出本次证明已经找到的与目标相关的规则和失败 fail 消息。函数主要对目标列表的第一个目标进行证明(行 3)，当输入目标为空时，算法证明成功返回(行 2)。proof_locate 函数输入一个规则头部角色，输出为知识库寻找启发式建议的的求证节点。算法根据 proof_locate 函数的输出采用两种证明方法：本地规则推演(行 10~19)，和远程证明调用(行 5~9)。

本地规则推演分两种情况进行证明，一种是当第一个目标就在知识库中时(行 10)，直接证明其它目标(行 11)，并返回证明中与目标相关的规则(行 12)。另一种情况是第一个目标不能直接得到，需匹配六条推理规则进行信任证明(行 13)，其中 Rules 和 P 为待匹配的变量。算法中将推理规则用(f | RE, PS, RS)数据结构代表的证明启发规则表示，意为如果要证某节点属于角色 f 或释放规则项 RE 定义的变量值，根据规则集 PS 和某条推理规则，还需证明该节点属于角色集 RS 中的角色。信任证明对所有能够匹配目标头部的推演规则进行试探(行 13)，将它的子目标实例化(行 14)并递归证明(行 15)，如果证明成功则继续证明其它目标(行 16)，当子目标和其它目标都证明成功时返回相关规则集(行 17)，否则任一目标证明失败都记录相关规则集(行 18)并尝试匹配下一条规则(行 13)。如果本地推演的两种情况证明都无法

成功，则返回相关规则集和 fail 消息(行 19)。

```
1 rule_set recursive_prove( list goals)
2    if(goals = [ ]) then return ⊥                            /*目标为空，证明成功，算法结束*/
3    [h, b] ← first(goals)                                    /*第一个目标，h 为头部，b 为尾部*/
4    server ← proof_locate(h)                                 /*证明启发式，该向谁求证*/
5    if (server)                                              /*远程证明*/
6        if (fail ∈ ( α ← rpn_client(server, [h, b])))        /*调用远程证明的本地代理*/
7            return α                                         /*失败，停止证明，返回规则集*/
8        β ← recursive_prove(rest(goals))                     /*继续证明其它目标*/
9        return α ∪ β                                         /*返回相关规则集*/
10   If([h, b] ∈ KB)                                          /*本地推理，目标在 KB 中*/
11       β ←recursive_prove(rest(goals))                      /*证明其它目标*/
12       return β ∪[h, b]                                     /*返回 KB 规则集*/
13   foreach (h, Rules, P) ∈ PH                               /*匹配推理规则，向前推演*/
14       p← instantiate (P, b)                                /*将子目标实例化*/
15       if (fail ∉ ( α ←recursive_prove(p)))                 /*对子目标进行证明*/
16           β ←recursive_prove(rest(goals))                  /*证明其它目标*/
17           if (fail ∉ β ) then return α ∪ β ∪Rules          /*证明成功，返回 KB 规则集*/
18       uncmpt_rs ← uncmpt_rs ∪ α ∪ β ∪Rules                 /*匹配失败，记录相关规则集*/
19   return uncmpt_rs ∪fail                                   /*失败，返回规则集和 fail 消息*/
```

图 3.5 DPN 分布式证明协商算法

proof_locate 函数的实现如图 3.6 左，如果某个 find 启发规则与目标头部的角色匹配，则返回启发规则建议的节点，否则返回为空。当 proof_locate 返回一个节点 server 时，recursive_prove 函数进行远程信任证明调用，算法首先调用本地代理 rpn_client(行 6)，本地代理向 server 发送请求，并等待 server 的返回消息。服务节点代理 rpn_server 的实现如图 3.6 右，算法调用 recursive_prove 函数对目标进行证明，并通过 negotiation_send 函数将证明结果包括相关规则集和是否成功的消息保护性的发送给请求节点。negotiation_send 函数根据信任证书释放规则将满足要求的规则以信任证书的形式发送出去，不满足要求的规则不能发送，这样就能提供对信任证书信息的保护。

```
1 address proof_locate(h)
2     foreach ((h′ , prin)∈ PH)
3         if (h′ =h)   return prin
4     return ⊥
```

```
1 rpn_server( client, q)
2     α ← recursive_prove (q)
3     negotiation_send(client, α)
```

图 3.6 远程证明调用相关算法

3.4.2 DPN 算法性质

在 DPN 算法中，证明和寻找启发式规则起了很重要的作用，我们假设系统的启发式规则都是正确的。DPN 算法的正确性可以由 RTP 语言推演规则的正确性得到保证。

定理 3.3（DPN 算法的正确性）：在信任协商系统中，如果 DPN 算法能够找到某结论 S 成立的证据，则结论 S 在系统中是正确的。

为了说明 DPN 算法的完整性，首先介绍两个引理。文献[61]提出信任图的概念，信任图中的节点表示角色，边表示角色之间的关系即角色规则，文章通过分析得到信任图的正确性和完整性引理 3.1。文献[59]提出信任分布式证明方法，并分析了带远程信任证明调用的分布式证明的算法，得出结论引理 3.2。

引理 3.1（信任图的正确性和完整性）：节点 A 属于角色 R（R←A）当且仅当在规则库 KB、节点 A 和规则 R 构成的信任图中有一条从节点 A 到角色 R 的路径。

引理 3.2：对于一个利用启发式规则的带远程调用的分布式信任证明算法，如果它能在一个信任知识库 *P* 的集中环境中证明某结论，则它也能在该知识库 *P* 的分布式环境中证明该结论。

文献[61]将信任证书搜索算法分为前向、后向和混合搜索三种类型，由于 DPN 算法是从目标角色向目标节点搜索相关规则，因此 DPN 算法也是一个后向搜索证明算法。由引理 3.1 可知，在系统的全局知识库 P 中，如果某结论 S：R←A 成立，则在 KB、R 和 A 构成的信任图中有一条从 A 到 R 的路径，也就是说 DPN 算法能够逆着该路径证明结论 S 成立。再由引理 3.2 可知，既然 DPN 算法能够在全局知识库 *P* 中证明 S 成立，那么在信任知识库分布式环境中，DPN 算法也能证明结论 S 成立。因此可以得到 DPN 算法的完整性定理。

定理 3.4（DPN 算法的完整性）：在信任协商系统中，如果结论 S 成立，则 DPN 算法能够找到证明结论 S 成立的证据。

3.5 实验分析

DPN 算法的特点是规则匹配的逻辑证明，因此我们采用 Amzi Prolog 逻辑语言来实现 DPN 算法，Prolog 语言本身的回溯原理和 Amzi 支持的网络通信使实现变得简单。为了对比分析 DPN 算法的性能，我们实现了 N.Li[55]基于 RT 语言的传统信任协商方法，采用后向搜索 Backward 算法，沿着信任角色定义节点收集信任证

书。实验模拟文中例 1 的应用，并增加用户的数量和角色的种类。根据角色的层次关系将所有角色分成三个层次如图 3.7 所示。一个书店(Org)向它的贵宾，信任客户和自己的员工提供不同的服务，书店的管理员(Reg)将一些服务授权给大学代理(Uni Agent)。书店(Org)、管理员(Reg)、大学代理(Uni Agent)以及每个客户分别运行自己的信任协商代理进程并拥有各自的局部知识库，代理程序包括 rpn_server 和 rpn_client。

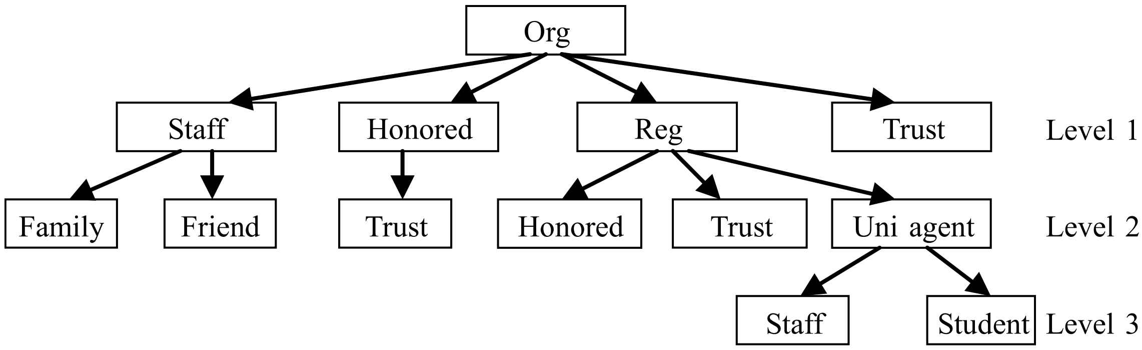

图 3.7 应用角色结构图

根据算法特点，两种算法分别在四种情况下进行性能比较： C1)任何节点都不缓存信任证书，C2)节点缓存第一层次的角色关系，C3) 节点缓存第一和第二层次的角色关系，C4) 节点缓存三个层次的所有角色关系。在缓存的角色关系中，一半为直接签名信任证书，另一半为逻辑签名信任证书。实验模拟 1000 个用户分别属于三层角色关系中的不同角色，他们向 Org 请求属于各自角色的服务。根据结点所属角色层次的不同，它们的服务请求分别用与其层次对应的 Q1，Q2 和 Q3 来表示。

实验在四种情况下对各层次的信任关系请求进行信任协商证明，统计两种算法的平均运行时间和节点间交互次数如图 3.8 和图 3.9 所示。在不缓存信任证书的情况下，两种算法对第一层次的信任请求问题有相同的交互次数，都为 3 次。信任请求问题每提高一个层次，Backward 算法交互次数增加 4 次，而 DPN 算法只增加 2 次。即信任请求层次越高，DPN 算法比 Backward 算法越高效。在缓存信任证书的情况下，缓存的角色关系每增加一层，DPN 算法交互次数减少 2。对 Backward 算法，当缓存第一层角色关系时，交互次数减少 1；当缓存更高层次的角色关系时，每增加一层缓存信任证书，交互次数减少 2。比较缓存信任证书导致两种算法交互次数减少的比例可以发现，缓存信任证书策略对 DPN 算法更有效。这是因为信任证明能更有效的利用缓存信任证书。

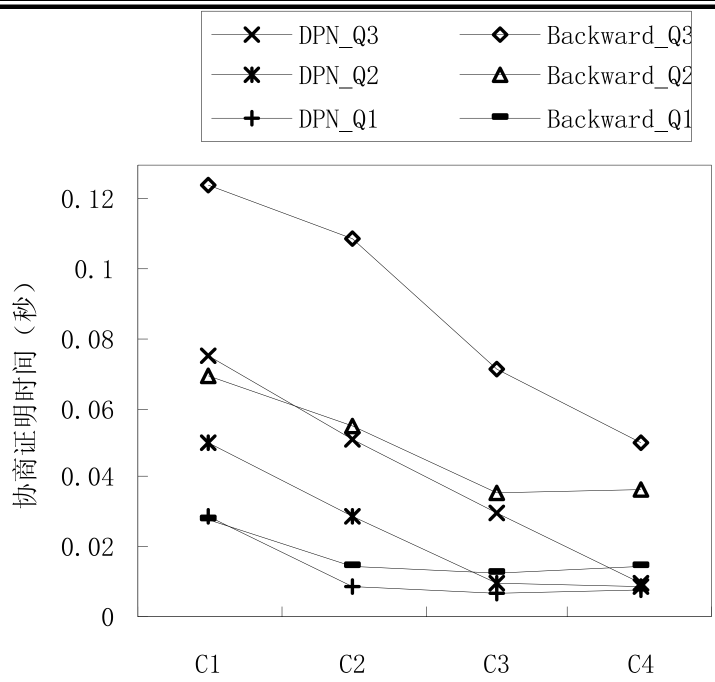

图 3.8 两种算法的运行时间

图 3.8 显示随着 DPN 算法比 Backward 算法交互次数的减少，算法时间也几乎成比例减少，最高减少 50%。这说明两种算法的运行效率主要由算法的交互次数决定。另外当两种算法交互次数相同的时候，Backward 算法的运行时间比 DPN 算法短，这是因为 DPN 算法需进行更多的规则匹配进行逻辑证明。由此可见信任证书收集算法应尽量减少规则匹配的调用。

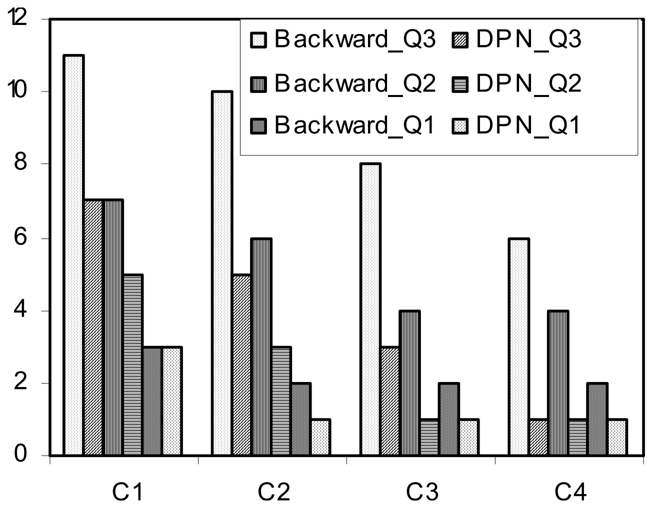

图 3.9 两种算法的交互次数

3.6 小结

作为跨安全域资源访问管理的主要方法之一，自动信任协商系统需能支持复

杂的访问控制和信任协商策略，并能高效的放置和搜集信任证书。本章借鉴已有研究成果提出面向信任分布式证明和协商的 RTP 语言，它可以支持信任分布式证明方法，定义复杂角色，保护信任证书敏感信息并避免信任证书盲目搜索。文章介绍了 RTP 语言的语法和语义以及利用它进行信任分布式证明协商的算法，实验表明该算法能够实现 RTP 语言的功能且比传统信任协商方法有很大性能提升。

当前很多应用需要在限制信任证书输出的情况下进行信任协商，本章的信任证书释放规则提供了信任证书输出限制的功能描述。下一步工作中，我们将在 RTP 语言和 DPN 算法的基础上，增加对信任证书加密技术支持，如隐藏信任证书和基于属性的加密技术等。另外信任的分布式证明方法在信任安全上是对传统信任协商方法的放松，保证信任证明的一致性成为信任协商系统很重要的因素，我们将开展这方面的研究。

第四章　基于行为信誉的信任量化和聚合计算模型

随着互联网和无线网络技术的迅速发展，面向服务的网络应用以及跨域的资源共享系统被广泛部署，并变得越来越重要。在这些动态开放的系统环境中，用户之间的陌生性往往导致服务消费者无法感知服务提供者的未来行为，这使得消费者必须承担更多的信任风险。为了减轻用户的风险从而提高系统的服务能力，提供一个可信的虚拟计算环境成为开放动态系统得以进一步发展的紧迫要求。由于信誉系统能够评估服务资源的可信性，基于信誉的信任管理正成为工业界和学术界的研究热点。信誉系统能够提供信任信息的聚合、过滤以及排序的功能，它在很多领域都被广泛应用，如分布式系统和应用[85,86,87,88,95,122]，多 Agent 系统[90,91]，基于 Web 的服务[92,93,119] 和推荐系统[84,106,110]等。

信誉是从历史行为记录中得到的一个关于信任概率的统计值。通常情况下，信誉值可以通过两种途径进行计算：根据评估者自身经历的直接信任评估和根据信誉反馈的间接信任评估[8]。为了能够计算直接信任，学者们提出了很多信誉模型，包括简单平均模型[83]、Bayesian 模型[94]以及基于证据的信念模型[90,91]]等，他们将信任量化成一个或几个确定性的估计值。尽管信誉本质上是一个概率的估计值，但现有的这些信任模型大多忽略了概率估计值的另一个重要属性：概率估计方差。信誉（可信概率）估计方差可以评估信誉估计值和真实信誉值之间的偏差，它在信誉系统中可以起到很重要的作用。例如，一个服务资源真实的事务成功率是 90%，一个服务消费者 A 与该服务资源进行了少量的事务交互，并根据自己的记录将该资源的信誉值确定为 70%。现有的信誉模型将不能支持评估者 A 对该信誉估计值的偏差进行评估，这将带来两个后果：(1)评估者不能确定自己给出的信誉估计值的准确性，因而也就不能确定该多大程度的依靠该信誉评估做出决策。(2)当评估者 A 将该信誉估计值作为反馈发送给其它信誉评估者 B 时，A 无法通告 B 应如何聚合该信誉反馈从而能得到更准确的信誉评估。

为了能够聚合信誉反馈，目前的大多数工作采用相加的聚合方法[85,86,88,108]，但是该方法很容易被恶意信誉反馈攻击[96]。为了解决这个问题，很多研究工作通过计算信誉反馈的可信度来检测恶意反馈，典型的反馈可信度计算方法包括基于距离的可信度[103]、基于信任值的可信度[86,108]以及基于个体相似性的可信度[85,96]。这些反馈可信度计算方法一般都假设评估者知道整个信誉系统的一些全局信任知识[85,96,108]，或者需要一些人为设定的参数[89,91]，这种强假设条件在实际应用中可能是无法实现的。我们认为这些基于相加方法的可信度检测技术之所以难以应用，原因在于相加的聚合方法缺乏对健壮性推测技术的支持。虽然相加的方法容易进

行反馈聚合，但这种感性的聚合方法没有统计推测理论的基础，容易被恶意节点控制。

为了评估和聚合信誉，本章介绍了一种健壮的线性马尔科夫 RLM（Robust Linear Markov）信誉模型。该模型的主要贡献包括：

1．RLM 模型将信誉评估表示成由信誉估计值和信誉估计方差组成的二维元组，并采用线性自回归方程定义信誉状态空间的演化，从而构成了一个隐马尔科夫过程。

2．模型采用卡尔曼滤波方法聚合信誉反馈，通过反馈噪音方差这个模型参数，卡尔曼聚合方法可以控制一个不正确反馈信誉值对模型的影响。该性质能够支持模型进一步采用健壮性的统计推测技术以抵御恶意反馈攻击。

3．设计了一个健壮性的模型校准方法。为了计算模型中的动态参数，模型首先采用 EM (Expectation Maximization)参数估计方法，它能自动产生适当的反馈噪音方差从而减轻一个恶意反馈信誉值的影响；在 EM 基础上，模型进一步采用基于假设检验的反馈检测方法，可以进一步抵抗恶意反馈的攻击。

RLM 信誉模型及卡尔曼反馈聚合方法完全基于统计推测理论。据我们所知，RLM 是第一个可以评估信誉估计方差的信誉模型，它能够给出更全面准确的信誉评估。另外，我们的卡尔曼反馈聚合方法以及模型校准方法能够抵御恶意反馈的攻击，并能自主地通过局部信任知识计算模型参数，无需人为设定参数。

本章第一节介绍相关工作，第二节描述 RLM 信誉模型，第三节介绍卡尔曼信誉反馈聚合方法，第四节描述 EM 参数校准以及基于假设检验的反馈检测方法，RLM 模型的实验分析在第五节，最后是本章的总结。

4.1 相关工作

信誉系统是一种基于群体反馈的的协同信任评估系统。为了评估一个资源的信誉，Song 等人[87]利用模糊逻辑理论来评估信誉，资源的信誉值是从一组规则中得到的信任的排序索引值。在 Bayesian 信誉系统[94]中，信誉值由统计结果的 Beta 概率分布函数计算。系统统计两个参数：总的正确结果次数 α 和总的错误结果次数 β。根据 Beta 概率分布函数，后验的信誉值可定义为 $\alpha+1/\alpha+\beta+2$。Wang 等人[90]将信誉定义成一个三维的信念元组(b, d, u)，其中三个参数分别表示出现正确结果、错误结果以及不确定结果的概率。综合分析上面三个信誉模型可以发现，他们都将信誉值量化为一个或几个确定的统计值。这些模型忽略了信誉估计值的另一个重要属性：信誉估计方差 [99]。在我们的 RLM 信誉模型中，信誉评估被直接定义为信誉估计值和信誉估计方差的组合，且 RLM 模型能同时对两者进行跟踪评

估。

为了能够聚合信誉反馈，最简单的办法是统计各信誉反馈中的正确结果数和错误结果数，这种简单相加的聚合方法很容易实现，被工业界广泛使用如 eBay[83]。在 Bayesian 信誉系统[94]中，信誉反馈包括两个参数：本次交互的正确结果数及错误结果数，它们也通过相加的方法被聚合到系统参数 α 和 β 中。因此我们可以认为 Bayesian 信誉聚合本质上讲也是一种相加的聚合方法。虽然相加的方法容易聚合反馈，但也容易被恶意信誉反馈攻击，且缺少对健壮性统计推测技术的支持。而本章采用卡尔曼反馈聚合方法，它能够通过反馈噪音方差这个参数控制信誉反馈对模型的影响，从而能支持进一步的健壮性统计推测技术以抵御恶意反馈攻击。

为了提升自己的信誉值或攻击其它合法节点的信誉，一个恶意节点很可能在信誉系统中提交恶意的反馈[120]。为此，很多研究工作利用反馈的可信度来决定反馈在聚合过程中的权值。在基于信任值的反馈可信度[84,86]方法中，来自某个节点的所有信誉反馈将拥有相同的可信度，它们由节点的全局信誉值决定。这种可信度方法在以下情况下将不能抵御恶意反馈：一个节点通过提供好的服务质量获得很高的信誉，但它却向系统发出很多关于自己竞争对手的恶意信誉反馈。在基于个体相似性的可信度测量 PSM (personalized similarity)[96] 中，为了评估来自节点 v 的信誉反馈的可信度，节点 W 计算它跟节点 v 关于系统中某些节点信誉评价的相似性。PSM 方法的缺点是节点 W 需要知道节点 v 关于某些系统节点的信誉评价，这种系统的全局信任信息在实际应用中有时很难获得。其它的可信度方法还包括 Yu 等人提出的 WMA (Weighted Majority Algorithm)权值方法，以及 Whitby 等人提出的恶意反馈分位数检测法。这两种方法需要人为地设定一些参数，而这在实际应用中很难确定。本章中，我们采用 EM 算法以及基于假设检验的反馈检验方法来共同抵抗恶意反馈攻击。另外，EM 算法可以自主地通过局部信任知识计算模型参数，无需人为设定参数。

4.2 RLM 信誉模型

基于信誉的信任系统一般包括系统结构和信誉计算模型两部分。系统结构主要解决如何存储和收集信誉反馈的问题，而信誉模型则用来表述并聚合信誉值。本章的目标是提出一种健壮的信誉计算模型。假设评估者可以不间断地接收到有关评估对象的信誉反馈，该评估者可以利用 RLM 信誉模型对评估对象进行信誉评估。

如果将信誉反馈看成是节点行为的观察取样，那么一个节点的信誉值本质上是该节点行为的概率统计估计值，因此我们采用统计值的形式来定义信誉评估。

假设一个节点的真实信誉值为 R，我们将该信誉的评估定义为一个二维的元组 $rep=\{\langle R\rangle,P\}$，$\langle R\rangle$ 是信誉的估计值，P 是该信誉估计值的估计方差，它表示信誉估计值 $\langle R\rangle$ 和信誉真实值 R 之间的平方差。同样，一个节点可以将自己对某个评估对象的信誉评估作为反馈发送给其它节点，因此一个信誉反馈也包括两个属性：反馈信誉值（信誉估计值）和反馈方差（信誉估计方差）。

为了维护一个节点的信誉，我们假设评估者可以不间断的接收到从其它节点发送过来的信誉反馈。在信誉反馈会话 k 中接收到的反馈表示为 $f_k=\{z_k,c_k\}$，z_k 和 c_k 分别表示反馈信誉值和反馈方差。在接收到反馈 f_k 后，评估者对节点的实时信誉值 R_k 进行估计，同时对该估计值的估计方差 P_k 进行评估。理想情况下，信誉的反馈值应该和信誉的真实值相等。但由于反馈推荐者往往不可能具有全部的系统信任信息，且服务具有抖动性的特点，因此推荐者给出的反馈信誉值和真实的信誉值之间往往都存在偏差。我们将这个偏差模型成一个均值为零的高斯噪音，噪音的方差为 Q_k，因此反馈信誉值和真实信誉值之间的关系可定义为：

$$z_k = R_k + q_k \qquad where \qquad q_k \sim Normal(0,Q_k) \tag{4.1}$$

对于一个正常的服务资源，我们假设其真实信誉值服从随机过程。在统计推测理论中，信誉值的估计属于一个无限脉冲响应的滤波问题，即信誉系统的输出值（新的信誉估计值）由系统的上一个输出（前次信誉估计值）及输入（信誉反馈）决定。为了解决这个问题，很多推测技术采用线性自回归模型来定义系统的输出，并具有较好的推测性能[99]。因此本章也采用线性自回归的模型来定义信誉状态的演化，而非线性演化则可以采用局部加权的方法来进行类似的处理[98]。作为一个一次近似处理，节点的真实信誉值 R_k 可以由一阶线性自回归方程定义为：

$$R_k = A_k R_{k-1} + w_k \quad where \;\; w_k \sim Normal(0,W_k) \tag{4.2}$$

A_k 是信誉状态转换因子，W_k 是信誉转换噪音方差。公式 4.1 和 4.2 定义了一个线性信誉状态空间，模型中各变量的关系如图 4.1 所示，其中圆圈节点表示信誉系统的目标变量，双圆圈节点表示观察到的信誉反馈，矩形节点表示需要校准的动态参数。由图 4.1 可以发现 RLM 信誉模型构成了一个线性隐马尔科夫过程，模型中系统状态 R_k 是一组未知的状态变量。我们的目标是从模型中得到信誉 R_k 的估计值，并评估出该估计值的估计方差 P_k，这将在下一节中通过卡尔曼反馈聚合方法得到。模型中的参数 A_k，Q_k 和 W_k 需动态确定，并能抵御恶意反馈攻击，这将在第四节中介绍。

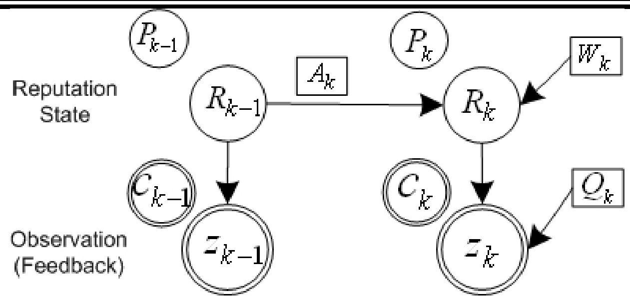

图 4.1 RLM 信誉模型图例

4.3 卡尔曼反馈聚合

在 RLM 信誉模型中，信誉状态的演化过程就是聚合信誉反馈得到信誉评估的过程。卡尔曼滤波方法是线性高斯系统的最优线性估计方法，它可以给出目标系统状态的最小均方差估计[97]。由于 RLM 信誉模型具有线性特征，因此本章采用卡尔曼滤波方法来跟踪信誉状态的演化并进行信誉反馈的聚合。卡尔曼反馈聚合方法可以给出信誉值 R_k 的最小均方差估计，并能够同时对该信誉估计值的方差 P_k 进行评估。更重要的是，由于卡尔曼聚合方法能够通过参数反馈噪音方差来控制一个反馈对模型的影响，它可以为健壮性统计推测技术提供支持，从而能抵抗恶意信誉反馈的攻击。

在运行卡尔曼反馈聚合之前，模型的动态参数 A_k, Q_k 以及 W_k 假设已知，它们将由下一节的健壮性模型校准方法确定。假设 R_k' 表示真实信誉值 R_k 的后验估计值，P_k' 是信誉估计方差 P_k 的后验估计值，符号<>表示估计算子。典型的卡尔曼滤波方法包含两个计算步骤：传播步骤和更新步骤。RLM 信誉模型相对应的两个卡尔曼反馈聚合步骤可以由公式 4.3~4.7 定义。

传播步骤

$$R_k' = A_k \langle R_{k-1} \rangle \tag{4.3}$$

$$P_k' = {A_k}^2 P_{k-1} + W_k \tag{4.4}$$

在传播步骤中，算法根据 RLM 模型计算出后验的信誉估计值 R_k' 和信誉估计方差 P_k'。为了能够运行卡尔曼反馈聚合方法，本章将信誉的初始估计值 $\langle R_0 \rangle$ 设为 0.5，这表示我们完全不知道评估对象的初始信任情况。信誉评估的初始方差 P_0 设为 0.01，这表示我们很不确定初始信誉估计值 $\langle R_0 \rangle$ 的准确性[99]。

更新步骤

$$S_k = P_k' + Q_k \tag{4.5}$$

$$\langle R_k \rangle = R_k' + \frac{P_k'}{S_k}(z_k - R_k') \tag{4.6}$$

$$P_k = \frac{Q_k}{S_k} P_k' \tag{4.7}$$

在更新步骤中，算法聚合反馈信誉值 z_k 以最小化信誉估计值的均方差。公式 4.5 计算出信誉估计值的残留方差 S_k。在公式 4.6 中，信誉的最终估计值将根据反馈信誉值的偏差 $(z_k - R_k')$ 以及信誉反馈方差和残留方差的比率 P_k'/S_k 来调整。从公式 4.5 中可以发现如果一个信誉反馈的噪音方差 Q_k 很大，那么 P_k'/S_k 将会很小，从而导致反馈信誉值只能在公式4.6中给信誉的最终估计值 $\langle R_k \rangle$ 带来很小的影响。因此我们可以得到以下定理：

定理 4.1 在 RLM 信誉模型中，如果信誉反馈 f_1 的噪音方差比信誉反馈 f_2 大，即 $Q_1 > Q_2$，那么在卡尔曼反馈聚合后，信誉反馈 f_1 对最终信誉估计值 $\langle R \rangle$ 的影响将比 f_2 小。

定理 4.1 说明在卡尔曼反馈聚合方法中，我们可以利用反馈噪音方差 Q_k 这个参数来减小恶意信誉反馈的实际影响。这个重要特性将使得 RLM 模型能够支持更进一步的技术扩展以抵抗恶意信誉反馈攻击。在公式 4.7 中，信誉的估计方差评估值 P_k 将根据信誉噪音方差和残留方差的比率 Q_k/S_k 来调整，我们可以得到下面关于信誉估计方差评估的定理。

定理 4.2 如果两个信誉反馈 f_1 和 f_2，它们有相同的信誉状态转换参数（$A_1 = A_2$ 且 $W_1 = W_2$），但反馈 f_1 有更大的反馈噪音方差（$Q_1 > Q_2$），那么在某个信誉状态下，当 RLM 模型分别聚合完这两个反馈后，反馈 f_1 将会导致新的信誉评估有更大的评估方差，即对于新的信誉评估，$P_1 > P_2$。

证明： 假设有两个信誉反馈 f_1 和 f_2，在模型的某个状态中，有 $A_1 = A_2, W_1 = W_2$ 但 $Q_1 > Q_2$。由公式 4.5 我们可以发现 $S_1 = S_2$，从而得到 $Q_1/S_1 > Q_2/S_2$，根据公式 4.7 可知，这将导致一个更大的信誉估计方差 P_1，因此结论得证。

4.4 健壮的 RLM 模型校准

在聚合信誉反馈并跟踪信誉状态演化之前，首先需要确定模型中的动态参数值 A_k，Q_k 和 W_k。另外，在模型参数的确定过程中，须考虑模型的健壮性，从而能够抵御恶意推荐的攻击。本节将首先介绍 EM (Expectation Maximization) 参数自动校准算法 ，对于一个有非法信誉值的反馈，该算法可以自动减少它对模型的影响。进一步，我们将介绍基于假设检验的反馈检验方法，它可以抵抗既有非法信誉值又有非法反馈方差的恶意反馈攻击。

4.4.1 模型参数校准

在统计推测理论中，模型的动态参数可以由 Expectation Maximization（EM）算法进行校准。EM 算法是一个最大似然估计方法，它可以给出参数的最小平方差估计。另外，EM 算法可以抵抗部分恶意信誉反馈的攻击。

对于模型中的参数，我们的目标是选择合适的参数值以便让系统输出值的可能性最大，即最大化信誉估计值序列的对数可能性 $\log p\left(R_{1:N}\right)$。根据数学分析的结果，该对数可能性的下限[82]可定义为：

$$\begin{aligned}\log p\left(R_{1:N}, \mathrm{z}_{1:\mathrm{N}}\right) = & \sum_{\mathrm{i}=1}^{\mathrm{N}} \log p\left(\mathrm{z}_{\mathrm{i}} \middle| R_{\mathrm{i}}\right) \\ & + \sum_{\mathrm{i}=1}^{\mathrm{N}} \log p\left(R_{\mathrm{i}} \middle| R_{\mathrm{i-1}}\right) + \log p\left(R_0\right)\end{aligned} \tag{4.8}$$

在信誉状态序列 $R_{1:N}$ 已知的情况下，数学方法很容易找到最大化上述对数可能性的参数值。但由于信誉状态序列在 RLM 模型中是未知的，信誉状态本身需求求解，因此公式 4.8 不能够直接通过数学方法求解。针对这种情况，本章采用 EM 参数估计方法。EM 算法把公式 4.8 的最大化问题转变成一个信誉状态假设已知的两步求解问题，它主要包括 Expectation 步骤和 Maximization 步骤。在 Expectation 步骤中，EM 计算出当前信誉状态的可能性分布；在 Maximization 步骤中，EM 方法进一步计算出各参数的最佳可能值，从而最大化信誉状态序列的可能性。

在 RLM 模型中，一个重要特征就是信誉反馈包含信誉反馈方差这个属性。该属性暗示了应如何聚合反馈以便评估者能够得到更全面准确的信誉评估。为了能够体现信誉反馈方差对模型计算的影响，本章对传统的 EM 方法进行扩展，增加初始化这一步骤。因此，每当评估者接收到一个信誉反馈 $f_k = \left\{z_k, c_k\right\}$，EM 算法将运行一次包含三个步骤的算法过程，算法的公式描述如下：

Initialization Step

$$Q_k = c_k,\ A_k = 1,\ W_k = \beta$$

Expectation Step

$$\Sigma_k = W_k^{-1} + Q_k^{-1} \tag{4.9}$$

$$\langle R_k \rangle = (W_k^{-1} A_k \langle R_{k-1} \rangle + Q_k^{-1} z_k) / \Sigma_k \tag{4.10}$$

Maximization Step

$$A_k = (\sum_{i=1}^{k} \langle R_i \rangle \langle R_{i-1} \rangle) / (\sum_{i=1}^{k} \langle R_{i-1} \rangle^2) \tag{4.11}$$

$$Q_k = \frac{1}{k} \sum_{i=1}^{k} (z_i - \langle R_i \rangle)^2 \tag{4.12}$$

$$W_k = \frac{1}{k} \sum_{i=1}^{k} (\langle R_i \rangle - A_i \langle R_{i-1} \rangle)^2 \tag{4.13}$$

在算法的初始化步骤中，信誉反馈的噪音方差设为信誉反馈方差 c_k。我们假设新的信誉状态跟前次信誉状态没有变化，因此信誉状态转换因子被初始化为 1。对于信誉转换噪音方差的初始值 β，它需根据评估者对信任环境的具体评估而定。当信任环境不稳定时，β 应该设为 0.01，否则应为 10−4[99]。在算法的 Expectation 步骤中，算法根据信誉值的条件分布计算出信誉的估计值 $\langle R_k \rangle$。在 Maximization 步骤中，算法根据反馈信誉值 z_k 和信誉估计值 $\langle R_k \rangle$ 的偏差选择动态参数值，以最大化信誉状态序列的可能性。

在 EM 算法中，通过以下定理，Maximization 步骤可以自动减轻一个恶意信誉反馈对 RLM 模型的影响。

定理 4.3 在 RLM 信誉模型中，如果一个恶意反馈只是非法改变它的反馈信誉值，而没有改变应有的反馈方差，那么与原有的正常信誉反馈相比，EM 参数校准算法将会使该恶意反馈对信誉模型的影响更小。

证明：对于一个恶意信誉反馈 f_k，假设它仅仅非法改变了原有的反馈信誉值 z_k，那么一般情况下，该反馈信誉值与信誉估计值 $\langle R_k \rangle$ 的偏差将会比原有的正常反馈信誉值大。而这个更大的偏差 $(z_k - \langle R_k \rangle)$ 将会导致 EM 算法给反馈 f_k 确定出一个更大的反馈噪音方差参数 Q_k（公式 4.12）。定理 4.1 说明一个信誉反馈的反馈噪音方差越大，它对最终信誉估计值的影响将越小。因此对于一个只非法改变反馈信誉值的信誉反馈，它对最终信誉评估的影响将会比一个正常信誉反馈小。

尽管 EM 算法可以通过定理 4.3 抵抗部分恶意反馈的攻击，但它仍存在脆弱性能够被恶意反馈攻击，该脆弱性可被描述为：

定理 4.4 在 RLM 信誉模型中，如果一个恶意反馈将它的反馈方差设为一个很低的值，那么无论该恶意反馈如何改变它的反馈信誉值，通过 EM 参数校准算法，它仍将会对信誉模型的信誉评估产生很大的影响。

证明：对于一个恶意信誉反馈 f_k，假设它有一个很低的反馈方差 c_k（接近 0）。在 EM 算法中，参数信誉噪音方差 Q_k 被初始化为 c_k。通过公式 4.10，一个很低的 c_k 和 Q_k 将会导致反馈信誉值 z_k 在信誉估计值 $\langle R_k \rangle$ 中占有很大比例。由于 z_k 和 $\langle R_k \rangle$ 存在这种强关联性关系，无论反馈信誉值 z_k 如何偏离真实信誉值 R_k，它和信誉估计值 $\langle R_k \rangle$ 的偏差 $(z_k - \langle R_k \rangle)$ 都将会很小，这将会导致公式 4.12 中的信誉噪音方差 Q_k 很小，从而根据定理 4.1，我们可以得到定理 4.4 中的结论。

4.4.2 恶意反馈检测

在上一小节中，我们介绍了 EM 算法来对模型中的动态参数进行校准。在抵抗恶意反馈方面，尽管 EM 算法可以通过调整反馈噪音方差这个参数来减少非法反馈信誉值对模型的影响，但该算法仍可能被恶意反馈攻击。当恶意反馈将反馈方差设为很低的值时，不管反馈信誉值如何恶意，该反馈仍可能获得对模型较大的影响力。而在这种情况下，我们的 RLM 信誉模型的性能将会下降。

为了增加模型的健壮性，本章在 EM 算法的基础上增加基于假设检验的恶意反馈检测方法。在第 4.3 节中，每当模型接收到一个信誉反馈 $f_k = \{z_k, c_k\}$，卡尔曼聚合方法将会给出一个信誉估计值 $\langle R_k \rangle$。针对原假设 H0：该信誉反馈是诚实的，反馈信誉值 z_k 和信誉估计值 $\langle R_k \rangle$ 的偏差应该服从一个均值为 0，方差为 $P_k + Q_k$ 的正态分布，其中 P_k 由卡尔曼聚合方法给出，Q_k 由 EM 算法决定。

为了能够检测出恶意信誉反馈，假设检验方法将评估反馈信誉值和信誉估计值之间的偏差是否超过正常范围。给定显著性水平 α，我们需要动态找到模型的检验阈值 t_K，它能够满足：

$$P\left(\left|z_k - \langle R_k \rangle\right| \geq t_k \middle| H_0\right) = \alpha . \tag{4.14}$$

当原假设 H0 为真时，$(z_k - \langle R_k \rangle)$ 服从一个均值为 0，方差为 $P_k + Q_k$ 的正态分布，因此我们可以得到：

$$P\left(\left|z_k - \left\langle R_k \right\rangle\right| \geq t_k \left| H_0 \right.\right) = 2 \times \theta\left(t_k \big/ \sqrt{P_k + Q_k}\right) \tag{4.15}$$

其中 $\theta(x)=1\text{-}\Phi(x)$，而 $\Phi(x)$ 是一个标准正态分布的累积分布函数 CDF(cumulative distribution function)。通过公式 4.14 和 4.15，我们解方程可以得到：

$$t_k = \sqrt{P_k + Q_k}\,\theta^{-1}\left(\alpha/2\right) \tag{4.16}$$

如果反馈信誉值 z_k 和信誉估计值 $\langle R_k \rangle$ 的偏差超过了阀值 t_K，则假设不成立。因此，该反馈将被认定为恶意反馈，而它所造成的信誉状态更新也将被放弃。某些情况下，一个正常的信誉反馈也可能被检测为恶意反馈，这被称为假阳性。尽管提高假设检验的显著性水平 α 可以增加真阳性（恶意反馈被检测出来），但它也会同时增加检测的假阳性。相关实验表明[97,101]，将显著性水平 α 设为 5%可以在真阳性和假阳性之间取得平衡，因此本章也将 α 设为 5%。

在上一小节的 EM 算法中，一个恶意反馈仍可以将反馈方差设为一个很低值，从而能够降低模型的性能。定理 4.5 将证明基于假设检验的反馈检测方法能够抵御这种攻击。

定理 4.5 在 RLM 信誉模型中，如果一个恶意反馈将它的反馈方差设为一个比原有值更低的值，那么它将变得更难通过本章的基于假设检验的恶意反馈检测方法。

证明：对于一个恶意信誉反馈 f_k，假设它给出一个比原有值更低的反馈方差 c_k。由定理 4.4 的证明我们可以发现，一个更低的 c_k 将会导致公式 4.12 产生一个更小的信誉噪音方差 Q_k。根据定理 4.2，更小的参数 Q_k 将会进一步导致公式 4.7 计算出一个更小的信誉估计方差预计值 P_k。综上，一个更小的 c_k 将会产生一个更小的 $P_k + Q_k$，而这将导致公式 4.16 计算出一个更小的检测阀值 t_K。因此，更小的检测阀值将会使反馈更难通过检测。

4.5 实验和结果

在本节中，我们将在一个模拟的计算环境中测试 RLM 信誉模型的性能。模拟的信誉计算环境包括正常的信誉反馈和恶意反馈。我们将进行三组实验来评估 RLM 模型的有效性、正确性和健壮性。

4.5.1 实验方法和指标

在本章的模拟实验中，每个信誉测试节点都跟系统中的其它节点进行 1000 次事务交互，因而关于每个测试节点我们都可以获得 1000 个信誉反馈，这将能充分测试信誉模型的评估性能。在每次事务交互中，测试节点的真实信誉值随机变化。我们将真实信誉值的变化范围设定在 0.6 和 1.4 之间，即节点本次交互的信誉值和上次交互的信誉值的变化不超过 40%。另外，我们将节点真实信誉值的最小和最大值分别设为 0.1 和 1。

在每次事务交互后，随着节点真实信誉值的变化，实验将模拟产生两种信誉反馈：正常反馈和恶意反馈。正常反馈反映一个诚实的信誉推荐者的行为。在现实应用中，由于不可能得到完全的信任知识，即使一个诚实的推荐者也不能给出完全正确的信誉反馈。因此实验中，正常反馈信誉值由真实的信誉值和一个均值为零的高斯噪音相加得到。该噪音的方差在实验中是一个动态的变量$k\sigma$，其中 k 是噪音方差的比例因子，σ是噪音的方差单位。实验中 k 的取值为 1、2 或者 3，从而能够测试不同的反馈噪音对 RLM 信誉模型影响。由于信誉反馈是一个主观的不确定性值，单位噪音方差σ被设为 0.01，这表示一个比较大的噪音[99]。

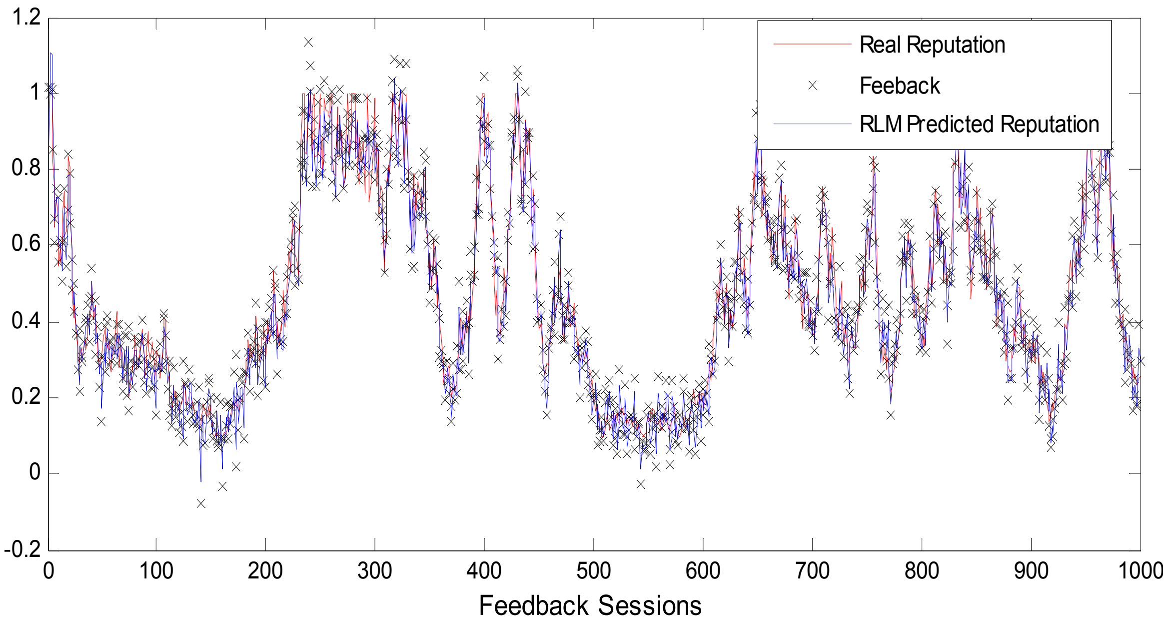

图 4.2 测试节点的真实信誉值、反馈信誉值和 RLM 模型给出的信誉估计值

在上文产生的所有正常反馈中，我们选取部分反馈将它们模拟成恶意反馈。实验中，恶意反馈的选取比例 p_m 是一个变量，它将被设为 10%，20%或者 30%，从而能测试不同比例的恶意反馈对模型的影响。恶意反馈可以分为两种类型：蓄意积极反馈和蓄意消极反馈[88,89]。对于一个被选择的正常反馈，如果它的信誉值低于 0.5，那么它将被改变为蓄意积极反馈，信誉值将变为区间[0.5, 1]中的某个随机值；否则它将被改变为蓄意消极反馈，信誉值将变为区间[0, 0.5]中的某个随机值。

为了能检测信誉模型的健壮性，我们将比较模型的以下指标：真阳（阴）性和假阳（阴）性。一个阳性反馈表示一个应该被模型拒绝的恶意反馈，而一个阴性反馈表示一个应被模型接受的正常反馈。假设模型中阳性和阴性反馈数目分别为n_p和n_n。一个假阳性反馈表示一个正常的反馈被模型错误的检测为恶意反馈，而一个真阳性反馈表示一个被正确检测出来的恶意反馈。假设模型中真阳性和假阳性反馈数目分别为n_{tp}和n_{fp}。假阳性率 FPR(false positive rate)表示被错误检测为阳性的反馈占所有正常反馈的比率，它可被定义为$FPR = n_{fp}/n_n$；同理，真阳性率 TPR(true positive rate) 表示被正确检测为阳性的反馈占所有恶意反馈的比率，它可被定义为$TPR = n_{tp}/n_p$。

4.5.2 RLM 模型的有效性

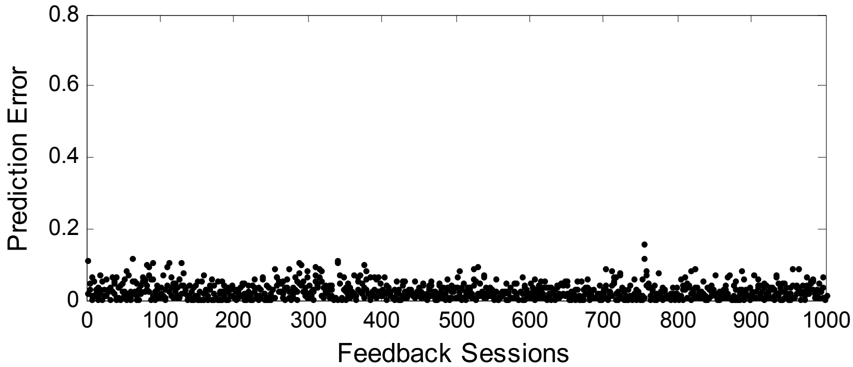

图 4.3 RLM 模型给出的信誉估计值和真实信誉值之间的误差

为了检测模型的有效性，本文将在一个没有恶意反馈的信誉环境中对 RLM 模型进行测试。当接收到一个新的信誉反馈，评估者使用 RLM 信誉模型计算评估对象的信誉估计值，并对该信誉估计值的方差进行评估。因此模型的有效性测试需检验 RLM 模型对信誉估计值和信誉估计方差两个属性的的表示能力。在第一组实验中，我们将信誉反馈的噪音方差设为 1σ，恶意反馈的概率 $p_m = 0$。图 4.2 是一个利用 RLM 模型对某节点信誉值进行估计的典型结果，红线表示该节点在每次事务交互中的真实信誉值，交叉符号表示带噪音的反馈信誉值，蓝线表示由 RLM 模型给出的信誉估计值。我们可以发现，尽管节点的真实信誉值在这 1000 个交互事务中的有很大的变化，而且信誉的反馈值也不准确，RLM 模型仍然能够给出很好的信誉评估。在大多数事务交互中，RLM 模型给出的信誉估计值和节点的真实信誉值都非常接近，以致代表它们的两条直线都重叠了。图 4.3 显示的是真实信誉值和 RLM 模型给出的信誉估计值之间的误差，可以发现绝大数的误差都小于 0.05。图 4.2 和图 4.3 都说明了 RLM 信誉模型能够很好的评估一个节点的信誉估计值。

RLM 信誉模型除了能给出信誉的估计值外，还能够同时对该信誉估计的方差 P 进行评估。我们将 RLM 模型评估出的信誉估计方差称为 RLM 估计方差（RLM estimated Prediction Variance）。同时，计算出模型给出的信誉估计值和真实信誉值之间的真实估计方差（Real Prediction Variance）。图 4.4 显示了这两种估计方差的关系，可以发现 RLM 信誉模型能够有效的评估信誉估计方差。在起始的 200 个反馈会话中，两种估计方差有较大出入，但在此后的大多数事务交互中，两种估计方差都很接近。这主要是因为 RLM 模型的很多变量都被初始化为常量，这需要模型运行一段时间才能消除它们的影响。

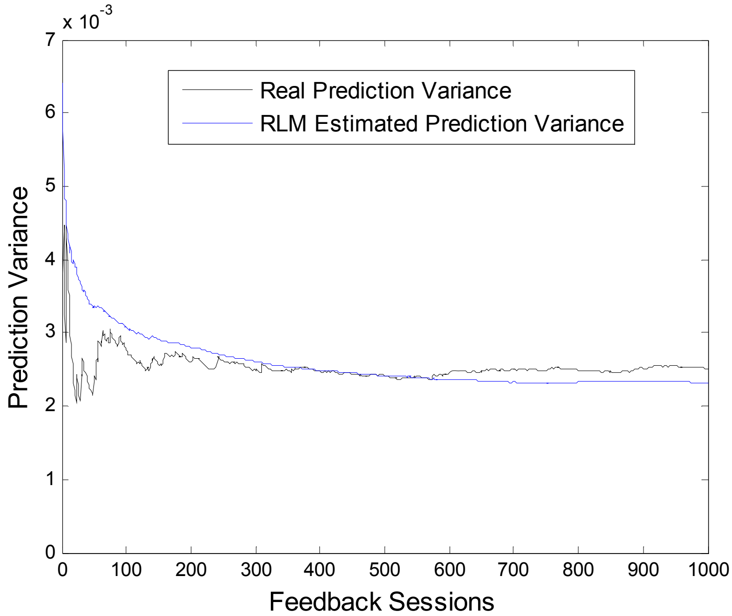

图 4.4 模型评估的和真实的估计方差

在上面的实验中，反馈的噪音方差被设为 1σ。下面，我们将进一步在不同的噪音方差（2σ 和 3σ）情况下测试 RLM 模型的性能。图 4.5 显示的是不同情况下，模型信誉估计值误差的累积分布 CDF(cumulative distribution function)。我们可以发现，在不同情况下，RLM 模型给出的大多数信誉值评估都比较准确。随着反馈噪音方差的增大，模型给出的评估误差也随之增大。这表明，反馈噪音方差的增大将会降低模型的性能。

4.5.3 RLM 模型的准确性

为了测试模型信誉评估的准确性，本文将比较 RLM 模型和相加模型[83]以及 Bayesian 模型[109]的信誉评估性能。信誉的相加模型能够很容易实现反馈聚合，并在商业服务领域（如 eBay）被广泛使用。通常情况下，相加模型通过平均各反馈信誉值得到最终的信誉值[8]。学术论文中还出现了很多加权的信誉相加模型，但在

本实验中，暂没考虑反馈的权值。在 Bayesian 信誉模型中，评估者统计评估对象总共的正确结果数 α 和错误结果数 β。根据 Beta 分布函数，最终的信誉值由公式 $\alpha+1/(\alpha+\beta+2)$ 确定。下面，本文将在没有恶意反馈的信誉环境中对模型的准确性进行两组实验。

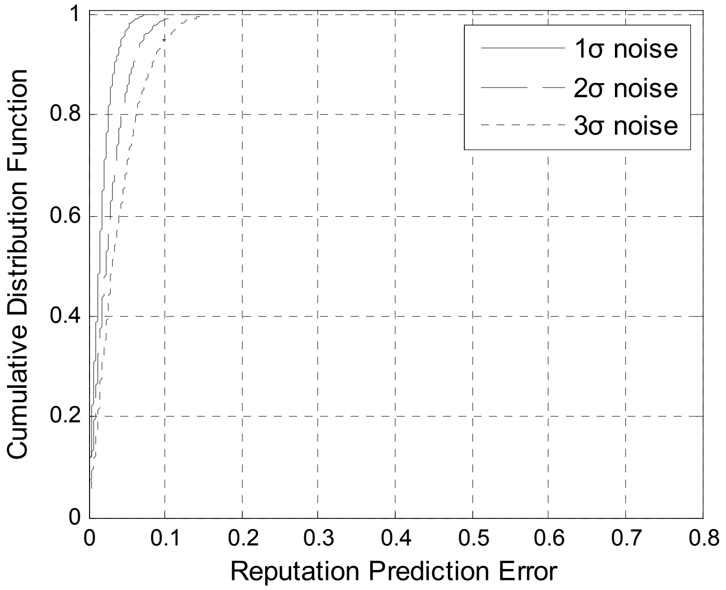

图 4.5 RLM 模型估计误差的累积分布 CDF

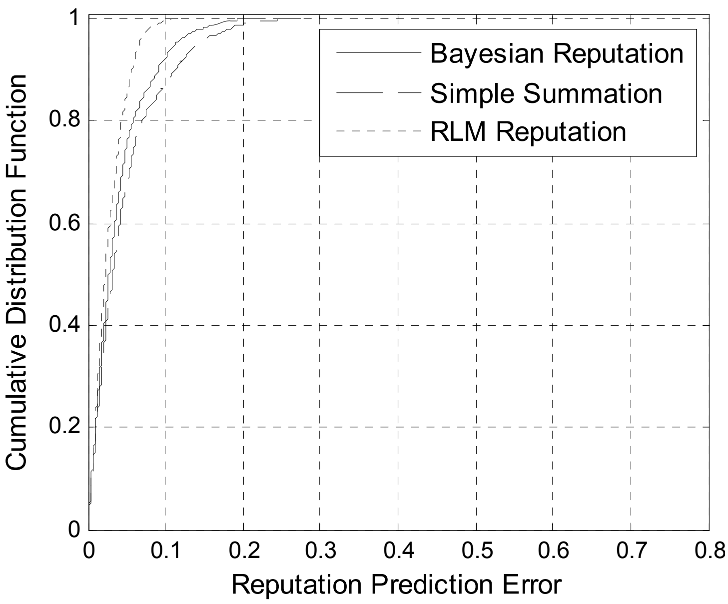

图 4.6 信誉估计误差的 CDF

实验首先将信誉反馈的噪音方差设为 1σ，三个信誉模型（simple summation, Bayesian 和 RLM）分别在相同的信誉反馈测试数据集上进行信誉评估。图 4.6 是三个模型给出的信誉评估误差的累积分布函数图 CDF。RLM 模型给出的大多数信

誉评估误差都小于 0.1，而另外两种模型的信誉评估误差分布在 0 到 0.2 之间。因此我们可以说在这三种信誉模型中，RLM 模型有最好的信誉评估准确性，Bayesian 模型的准确性比简单相加模型要高。图 4.7 显示的是三种模型给出的归一化均方误差 NMSE (normalized mean squared error)。该归一化均方误差可被定义为模型所有信誉值估计误差的均方差除以真实信誉的方差。图 4.7 证实了图 4.6 结果的正确性，即 RLM 模型有最好的信誉评估准确性，Bayesian 模型的准确性比简单相加模型要高。

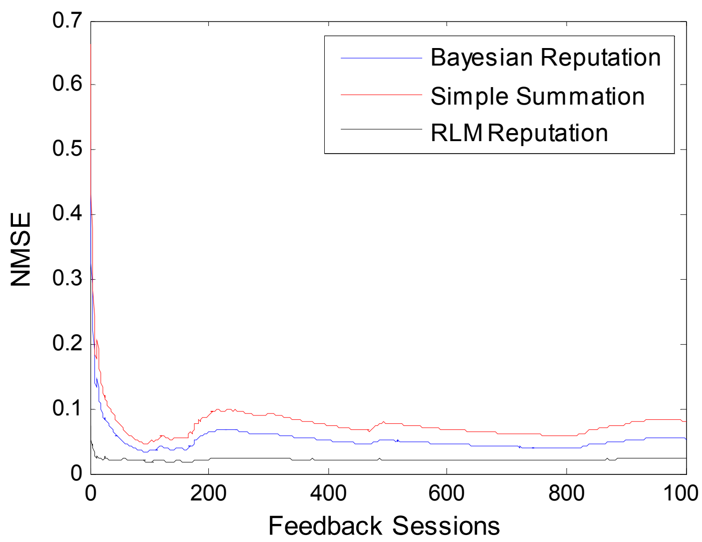

图 4.7 信誉模型的 NMSE

在第二个实验中，我们将信誉反馈的噪音方差分别设为 1σ，2σ 和 3σ。实验计算出真实信誉值和三个模型给出的信誉估计值之间的估计方差。由于 Bayesian 模型的准确性比简单相加模型要高，图 4.8 主要对 Bayesian 模型和 RLM 模型的信誉估计方差进行比较。在各种噪音设置情况下，RLM 模型给出的信誉估计方差都比 Bayesian 模型小。尤其当反馈噪音较小的时候，RLM 模型较 Bayesian 模型有显著的性能优势；当反馈噪音较大的时候，RLM 模型的性能优势将减小。这主要因为 RLM 模型使用最大似然估计的方法进行参数校准，该方法较易受反馈噪音的影响。因此而当反馈噪音较大的时候，RLM 模型的性能优势将会退化。图 4.9 显示的只是各种信誉模型在一个数据集上运行的结果，下面我们对实验进行扩展，在各种反馈噪音情况下，各信誉模型分别在五个数据集上进行实验。图 4.9 显示的是三种信誉模型在各种情况下的平均估计方差，它印证了图 4.8 得到的结果，即在这三种信誉模型中，RLM 模型可以给出最准确的信誉值评估，但当反馈噪音较大的时候，LHM 的性能优势将会缩小。

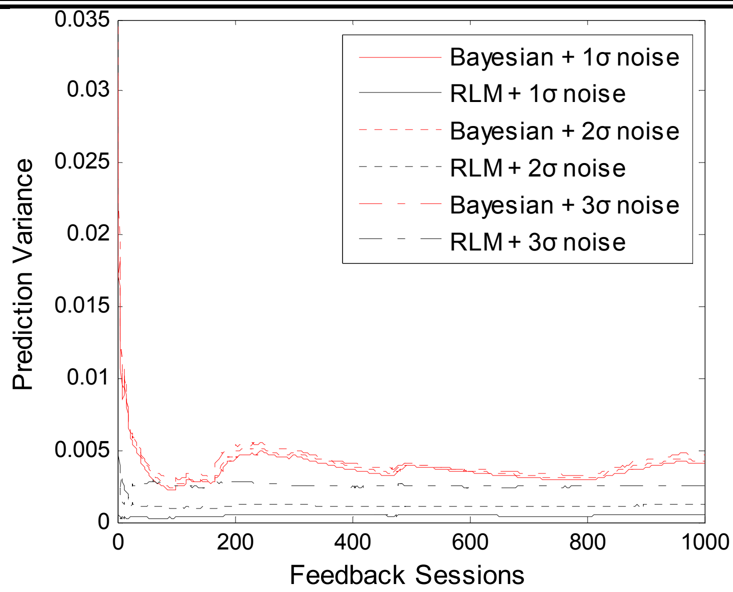

图 4.8 不同信誉反馈噪音情况下信誉模型的估计方差

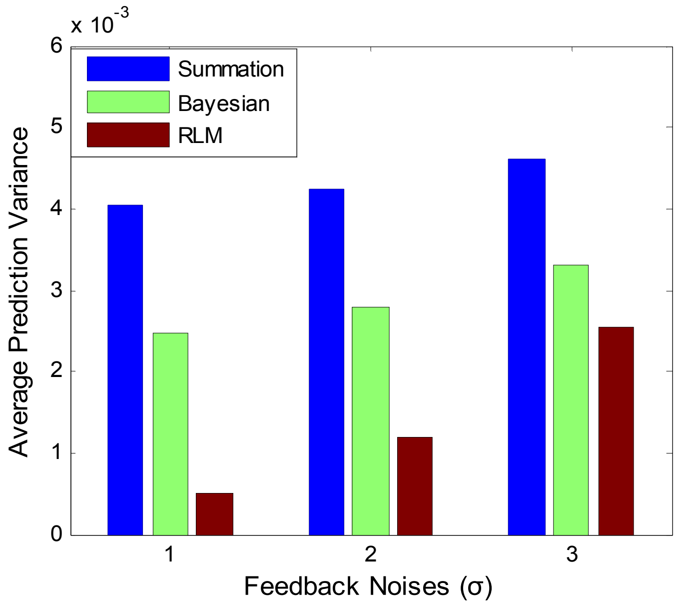

图 4.9 不同信誉反馈噪音情况下信誉模型的平均估计方差

4.5.4 RLM 模型的健壮性

在上面两小节中，文章在一个没有恶意反馈的环境中检测了 RLM 模型的有效性和正确性。下面我们将评估模型在恶意反馈攻击情况下的健壮性。为了抵抗恶意反馈，Whitby[89]等人在 Bayesian 模型的基础上提出了恶意反馈的分位数检测法（quantile filtering）。依据信誉的 Beta 分布函数，如果一个反馈信誉值落在该函数的 q 或者 $(1-q)$分位线外，那么它将被检测为恶意反馈。下文中带分位数检测

法的 Bayesian 信誉模型被简写成（Bayesian + Quantile）模型。根据文献[89]的实验结果，我们将分位数参数设为 0.01。为了能够比较模型的性能，本文还将测试没有假设检验方法的 RLM 模型性能，并将它称为 LM 信誉模型。

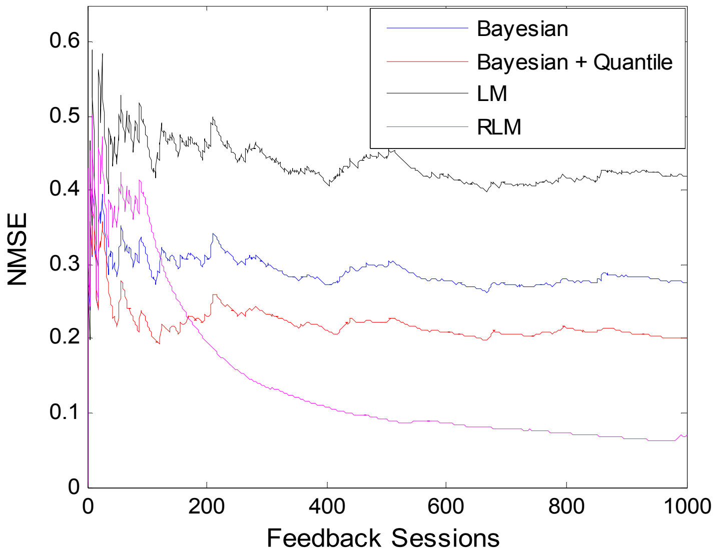

图 4.10 恶意反馈情况下各信誉模型的 NMSE

实验首先将恶意反馈比例 p_m 设为 20%，并在相同的反馈测试数据集上运行 Bayesian 模型、Bayesian + Quantile 模型、LM 模型以及 RLM 信誉模型。对于 LM 和 RLM 信誉模型，恶意反馈的反馈方差都被设为一个很小的值 10^{-4}，从而使每个恶意反馈都能对模型产生影响。图 4.10 给出四种模型的信誉值估计误差的归一化均方差 NMSE。在初始的 100 个反馈会话中，四个模型的性能都不稳定。随后，RLM 模型的性能逐渐提高，并最终给出最小的 NMSE。在这四个模型中，LM 模型的性能最差。这表明当存在反馈方差很低的恶意反馈时，没有假设检验方法支持的 RLM 模型将会非常脆弱。同时我们也可以看出带分位数检测法的 Bayesian 模型比纯 Bayesian 模型的性能更好。

下面我们进一步将恶意反馈的比例 p_m 分别设为 10%，20%和 30%，恶意反馈的反馈方差依然被设为一个很小的值 10^{-4}。在每种反馈比例 p_m 情况下，实验产生 5 个反馈测试数据集，从而可以得到有代表性的每种情况的平均结果。图 4.11 给出了四种模型的平均估计方差，它也印证了图 4.10 中的结论。除此之外，我们可以发现随着恶意反馈比例的增大，Bayesian 和 LM 模型的性能退化速度很快；当它们增加了恶意反馈检测方法后，对应的 Bayesian + Quantile 和 RLM 模型的性能退化速度相对较慢。另外我们也可以发现当比例 p_m 接近 30%时，所有四个信誉模型的性能都有很大的退化。

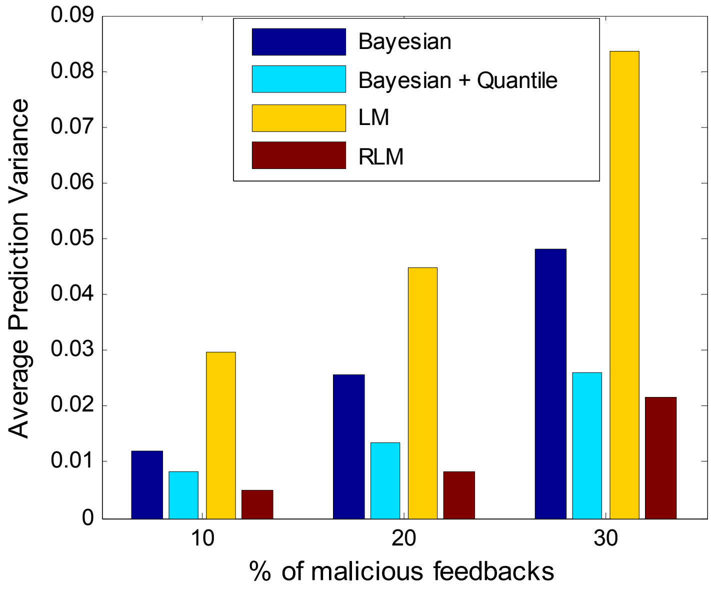

图 4.11 恶意反馈情况下信誉模型的平均估计偏差

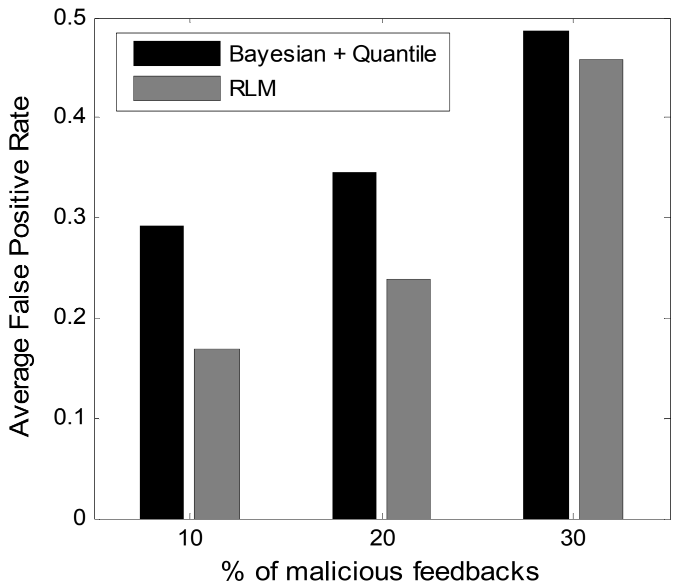

图 4.12 恶意反馈情况下模型的假阳性

Bayesian + Quantile 和 RLM 模型都能检测恶意信誉反馈，因此我们将通过真阳性率和假阳性率两个指标来比较它们的检测性能。图 4.12 和 13 显示，在各种恶意反馈比例情况下，RLM 模型都有着更高的真阳性率和更低的假阳性率。而随着恶意反馈比例的增大，两种模型的假阳性率都随之升高，而真阳性率将随之降低。这说明随着恶意反馈比例的增大，检测方法将会更难检测出恶意反馈。另外，我们也可以发现当恶意反馈比例 p_m 接近 30%时，两种检测方法的性能将会显著降低。大概只有一半的恶意反馈能够被检测出来，而被检出来的恶意反馈中，也有一半是被错误地标为恶意反馈，即它们应该是正常的反馈。

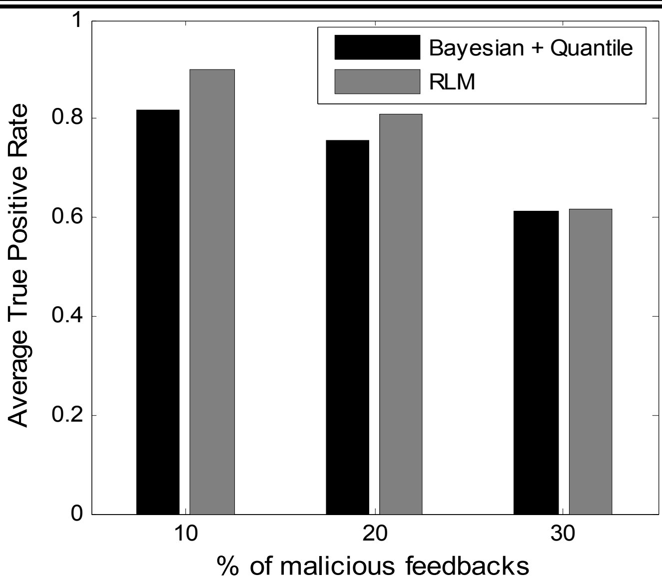

图 4.13 恶意反馈情况下模型的真阳性

4.6 小结

针对目前信誉模型不能评估信誉估计方差以及相加的反馈聚合方法难以支持健壮性信誉评估两个问题，本章描述了一个健壮的线性马尔科夫信誉模型 RLM（Robust Linear Markov）。RLM 模型将信誉评估表示成信誉估计值和信誉估计方差两个属性，并采用线性自回归方程定义信誉状态空间的演化。由于 RLM 模型构成了一个隐马尔科夫过程，模型采用完全基于统计推测理论的卡尔曼滤波方法，它通过参数反馈噪音方差为健壮性的统计推测技术提供了支持。为了给出一个健壮的模型参数校准从而能抵抗恶意反馈攻击，模型首先采用 EM 参数动态校准算法，该算法可以自动给出合适的参数选择，从而能减轻不正确信誉反馈值对模型的影响；文章还进一步在模型中引入基于假设检验的反馈检测方法，从而能够过滤恶意反馈。实验结果表明 RLM 信誉模型能够有效地跟踪评估信誉的估计值和信誉估计方差。与简单相加模型和 Bayesian 模型相比，RLM 模型能够给出更准确的信誉值评估。在恶意反馈攻击的情况下，RLM 模型在被测试的模型中能够给出最小的信誉值估计误差；且在恶意反馈检测方面，RLM 模型比 Bayesian + Quantile 模型有着更小的假阳性率和更高的真阳性率。

下一步工作中，我们将主要考虑如何减小信誉反馈噪音对 RLM 信誉模型的影响。目前，反馈的噪音将会导致模型性能的退化。此外，我们将基于现有的统计推测理论，在 RLM 信誉模型的基础上增加健壮性的信誉评估技术，从而能够防范恶意信誉反馈攻击。

第五章 基于身份策略和行为信誉的混合信任管理

开放网络环境下客户之间是陌生的，传统的基于角色策略的访问控制[47]无法对陌生客户进行访问控制。为此，Blaze[18] 等人首次提出了“信任管理”的概念，实现开放环境中的信任评估。现有的安全技术，无论是底层的密码算法和通信协议，还是高层的安全模型和策略，都隐含地与信任相关。它们的实现都依赖于某些信任假定前提，或者它们的目标就是为了获得并创建某种信任关系。因此，作为网络安全技术的重要基础和关键目标，系统的信任关系成为了网络安全研究的热点。

针对大规模的动态开放系统环境，很多研究者采用基于角色的凭证策略对信任关系进行定义和管理[55,65,66]。它们将信任定义为一个客观的逻辑关系，可以抽象为 0 或 1，也可以用布尔值表示。但在实际的应用中，大多数情况下用户希望能够实现更细粒度的信任管理，比如该多大程度地相信一个经济分析师的投资推荐，而不仅仅是相信或者不相信。另外，目前大多数基于角色的凭证管理技术是一种静态的信任管理技术，它不能跟踪角色的授权行为并动态管理角色关系，这将导致角色域内的用户行为无法被跟踪控制，并为角色域内的恶意行为提供了条件。

本章介绍了一个基于身份策略和信誉的混合信任管理系统 RTE (Role-based Trust Evaluation)。RTE 的信任策略语言通过在基于角色的信任关系语言中增加信誉值参数，从而能够在信任策略语言中支持信誉的管理。RTE 的信誉值计算包括信任经验和信任推荐，能够实现资源的细粒度访问控制授权。另外，RTE 的策略语言通过定义信任合成算子，能够支持信誉值的网络传递和计算，进而可以根据角色的跟踪记录，动态管理角色授权，抗击角色域内的恶意行为。文章给出了 RTE 策略语言的语法和推演规则，介绍了 RTE 系统的信任值计算，并给出了一个 RTE 系统进行混合信任管理的示例。

本章第一节介绍相关工作，第二节给出 RTE 系统策略语言的语法及其推演规则，第三节介绍 RTE 中的信任计算，第四节介绍一个 RTE 系统进行资源访问控制的实例，最后一节总结本章及介绍未来工作。

5.1 相关工作

Sandh[47]等人系统的提出基于角色的访问控制模型 RBAC (Role Based Access Control)，RBAC 被广泛使用，之后基于角色的策略语言发展迅速，Hayton[29] 等人提出一种角色定义语言 RDL (Role Definition Language)，RDL 通过描述用户获得角色的前提条件来对用户进行访问控制。信任管理[18] 提出之后，策略语言不但用于描述安全策略，还用于描述安全凭证。Li NH[55,65,66] 等人结合基于角色的访问控制

思想，对传统的信任管理框架进行了扩充，提出了基于角色的信任管理框架 RT (Role-based Trust-management)。

文献[55]中介绍了 RT_0 策略语言，RT_0 能够描述角色的本地授权、角色的层次关系、基于角色的委托授权、基于属性的委托授权及角色的交运算。文献[65]对 RT_0 策略语言进行扩展，衍生出 RT 系列语言，其中 RT_1 在 RT_0 基础上支持角色的参数化，使得一系列具有相似属性的角色可通过带参数的角色进行统一描述。而 RT_2 则又在 RT_1 的基础上加入了逻辑对象的概念，即对具有逻辑相关的对象，如系统资源、访问控制模型等进行分类，使角色的逻辑权限描述更简洁明了。RT^T 和 RT^D 是两个具有独立性质的描述语言，可以同时或单独与 RT_0、RT_1 和 RT_2 组合使用，其组合后的形式可写为 RT_i、RT_i^D、RT_i^T 或 RT_i^{DT}，其中($i = 0; 1; 2$)。RT_T 提供了簇角色和角色生成的操作子两类语言设施，主要用于表达带阀值的本地策略和安全责任分离的本地策略。RT_D 提供了基于角色触发的委托，用于表达能力的选择性使用和委托。

传统的 RBAC 是一种经典的访问控制模型，能够有效的控制系统内用户的访问，但是 RBAC 不支持对陌生客户访问的控制，RDL 只能够定义布尔类型的角色关系，不支持更细粒度的角色定义，RT 系列语言功能强大，易于使用，但同样只能支持布尔类型的角色定义，并且不支持信誉管理，不易于动态管理角色。本章在 RT 语言基础上进行扩展，提出一种支持信誉管理及细粒度角色定义，能够基于信任值动态管理角色的策略语言。

5.2 RTE 策略语言

本节首先介绍 RTE 系统中策略语言所需描述知识的框架，然后介绍 RTE 策略语言的语法，并给出语言的推导规则。

5.2.1 知识库框架

在一个信任域中，管理者能够对信任信息进行签名或收集其他系统的信任信息，网络中所有的节点构成一个集合，每个节点 A(Principal)都有一个局部知识库 KB(Knowledge Base)，记为 P_A，我们设计的 KB(如图 5.1 所示)包括四个部分：信任计算规则 TCP，服务访问控制规则 SGP，信任信息集合 TIS、信任计算结果集合 CRS。各部分介绍如下：

- 信任计算规则定义信任值在计算过程中的转化规则；
- 服务访问控制规则是本节点定义的基于角色的访问控制规则；

- 信任信息集合包含从本节点和其它节点收集的信任信息；
- 信任计算结果集合包含本节点通过信任计算所得的结果。

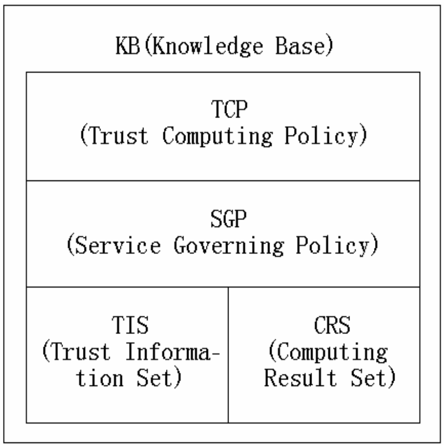

图 5.1 RTE 系统的策略语言知识库框架

5.2.2 策略语言的语法

图 5.2 给出了 RTE 系统的策略语言的 BNF 描述，下面解释中出现的数字指的是图中该数字行对应的语法。RTE 策略语言由两种类型的规则组成：信任计算规则(1)和服务访问规则(3)。

1) 信任计算规则

信任计算规则(1)由头部和尾部组成，头部和尾部以箭头“⇐”连接，头部为一个角色 role(6)，尾部(2)为若干个常量“*”角色之间以“⊕”进行的连接，表示对头部角色 role 的信任计算公式。“*”运算符表示赋权值，“⊕”运算符表示进行权值运算(计算方法将在 5.3 节中介绍)。

角色(6)为某个节点 principal 的一个角色项或角色项的连接(7)。角色项的连接由角色项之间通过点符号“.”连接而成。

2) 服务访问规则

服务访问规则(3)，规则头部为一个角色 role，箭头“$\overset{exp}{\leftarrow}$”表示经验，箭头“$\overset{rec}{\leftarrow}$”表示推荐，规则尾部(4)有两种类型，一种只有一个节点 principal，表示该节点属于规则头部的角色；另一种形式为 R1∩…∩Rn ，由一组角色组合而成，如果某节点符合规则尾部的要求，则该节点属于规则头部的角色。第二种形式的服务访问规则省略了一个变量，等同于(R←X)←(R1←X)∩…∩(Rn←X)。

角色项可带参数(11)和信任值(12)，参数和信任值以“;”分隔，比如 A.modify(File;?value ＞ 0.8)，A 是节点，modify 是操作名，参数 File 代表操作对

象，?value 是信任值，?表示变量，>是变量关系，0.8 是域值。

<policy-tc> ::= <role> "⇐" <integ of role> 1

<integ of role> ::= <constant> "*" <role> |
<constant> "*" <role> " ⊕ " <integ of role> 2

<policy-sg> ::= <role> <arrow> <body> 3

<body> ::= <principal> | <conj of role> 4

<conj of role> ::= <role> | <role> "∩" <conj of role> 5

<role> ::= <principal> "." <role-terms> 6

<role-terms> ::= <role-term> | <role-term>"." <role-terms> 7

<role-term> ::= <role-term1> | <role-term2> 8

<role-term1> ::= <role-name> | <role-name> "(" <parameters> ")" 9

<role-term2> ::= <role-name> "(;" <trust-field> ")" |
<role-name> "(" <parameters> ";" <trust-field>")" 10

<parameters> ::= <parameter> | <parameter> "," <parameters> 11

<trust-field> ::= <constant> | <trust-value> <relation> <constant> 12

<arrow> ::= "←" | "$\xleftarrow{\text{exp}}$" | "$\xleftarrow{\text{rec}}$" 13

图 5.2 RTE 语法的 BNF 描述

5.2.3 策略语言的推演规则

1) 实例推演 VI(variable instantiation)。如果规则$\phi \in P_A$，P_A为节点 A 的知识库，x 为ϕ中一个变量，b 为该变量定义域上一个常量，则 $P_A \mapsto sub_b^x\phi$， $sub_b^x\phi$ 表示用 b 替代 ϕ 中所有 x 的出现。若 ϕ="$f \leftarrow f_1 \wedge \ldots \wedge f_m$"，$f$, $f_1, \ldots, f_m$ 为角色，则 $sub_b^x\phi$ ="$(f \leftarrow b) \leftarrow (f_1 \leftarrow b) \wedge \ldots \wedge (f_m \leftarrow b)$"。

2) 角色连接 RL(role link)。如果规则ϕ="$A.R_1 \leftarrow B$"且$\phi \in P_A$，则 $P_A \mapsto \phi'$，ϕ'="$A.R_1.R_2 \leftarrow B.R_2$"。

3) 信任合成 1(Trust Integration1)：如果规则 $\text{A.R}^{\text{rec}}(;?\text{value}_1) \leftarrow \text{X} \in \text{P}_\text{A}$ 且 $\text{X.R}^{\text{rec}}(;?\text{value}_2) \leftarrow \text{X} \in \text{P}_\text{A}$，其中 X 为 P_A 中出现的任意节点，则 $\text{P}_\text{A} \mapsto \phi$ ，$\phi = \text{A.R}(;?\text{value}{=}\sum_{\oplus} ?\text{value}_1 * ?\text{value}_2) \xleftarrow{\text{rec}} \text{B}$ 。$\sum_{\oplus} ?\text{value}_1 * ?\text{value}_2)$ 表示对节点 X 以 $?\text{value}_1$ 作为 $?\text{value}_2$ 的权值进行信任值运算。

4) 信任合成 2(Trust Integration2)：如果规则 $A.R(;?value_1)\overset{rec}{\leftarrow} B \in P_A$ 且 $A.R(;?value_2)\overset{exp}{\leftarrow} B \in P_A$，则 $P_A \mapsto \phi$ ，$\phi = A.R(;?value=\alpha * ?value_1 \oplus \beta * ?value_2) \leftarrow B$ 。其中α、β为用户预先定义的常量。

5.3 RTE 中的信任计算

RTE 中的信任计算包括三个类型：经验计算，推荐计算和信任合成。本章规定信任值的值域为[-1,1]∪⊥，负值表示不可信，正值表示可信，0 表示介于可信与不可信之间，⊥是一个中立的值，表示不能对信任进行评估(比如之前没有过任何历史记录就不能通过经验计算判断用户可信性)。

1) 经验计算

当对本系统保留的历史数据进行分析时，应考虑到不同时间的历史经验对信任评估的作用是不同的(最近一段时间的经验自然比很久以前的经验更具权威性)，所以可将时间分为时间片，$t=[t_0,t_1]\cup[t_1,t_2]\cup\cdots\cup[t_n,t_{n+1}]$，新近的经验比较重要，时间片的划分粒度应细一些，即时间片的长度相对较小，久远的经验重要性较小，故划分粒度可粗一些，即时间片的长度相对较大。为体现时间对信任评估的影响，需为不同时间片分配不同的权值。经验计算如公式 5.1 所示：

$$T_a^{exp} = \begin{cases} \perp, & \text{if } \neg\exists a \in t \\ \dfrac{\sum_{i=0}^{n} w_i \sum_{k=1}^{n_i} v_k^i}{\sum_{i=0}^{n} w_i \sum_{k=1}^{n_i} \left|v_k^i\right|}, & otherwise \end{cases} \tag{5.1}$$

a 代表行为，T_a^{exp} 代表通过历史经验对行为进行计算而得的信任值，v_k^i 代表在时间片 i 中 a 第 k 个记录的表现值(良好为正值，恶劣为负值，取值范围为[-1,1])，w_i 代表时间片 i 的权值。如果在时间 t 内没有 a 的历史记录($\text{if } \neg\exists a \in t$)，那么 T_a^{exp} 取中立的值⊥，否则经过计算得出 T_a^{exp}，显然 T_a^{exp} 的值域为[-1,1]，正值表示可信，负值表示不可信。

2) 推荐计算

属性、行为均可由其他实体推荐，需要注意的是不同实体推荐的可信程度不同，推荐作为一个特殊的行为，它的可信度可由历史经验计算而得，并将作为实体所推荐的信任值的权值，信任计算公式如下：

$$T_a^{rec}=\frac{\sum_{i=1}^{n}(A\xrightarrow{rec}i)^a\cdot v_i^a}{\sum_{i=1}^{n}(A\xrightarrow{rec}i)^a}\quad\left(\left((A\xrightarrow{rec}i)^a\right)>0\right) \tag{5.2}$$

a 代表所推荐的属性或行为，T_a^{rec} 代表通过推荐对 a 进行计算而得的信任值，$(A\xrightarrow{rec}i)^a$ 代表对第 i 个实体推荐 a 的信任值，如果 $(A\xrightarrow{rec}i)^a<0$，表明 i 的推荐不可信，将不对其计算。$v_i^a$ 代表第 i 个实体推荐的 a 的信任值。由于对推荐这个行为进行了记录、分析及计算，所以在一定程度上抗击了恶意推荐行为。

3) 信任合成

综合经验与推荐可得行为的信任评估结果，，信任计算如公式 5.3 所示：

$$T_a=\alpha\cdot T_a^{exp}+\beta\cdot T_a^{rec} \tag{5.3}$$

T_a 代表 a 的综合信任值，可以对 T_a^{exp} 和 T_a^{rec} 赋不同权值 α 和 β，其中 α + β =1。

当对 a 进行计算时，可以为相关的 a_n 赋权值，进行计算，以此来推测 *a* 的可信程度。信任计算如公式 5.4 所示：

$$T_a=\frac{\omega_1\cdot T_{a_1}+\omega_2\cdot T_{a_2}+\cdots\omega_n\cdot T_{a_n}}{\omega_1+\omega_2+\cdots+\omega_n} \tag{5.4}$$

T_a 代表 a 的信任值，$T_{a_1},T_{a_2}\cdots T_{a_n}$ 为 $a_1,a_2\ldots a_n$ 的信任值，ω_1, ω_2 …ω_n 为用户自定义的权值。当参与信任计算的是证书等客观凭证，默认其信任值为最大值(正1)。

5.4 RTE 资源访问控制实例

需求：一家公司 A 有一个文件 File，A 希望他的项目组 1 的成员和他的合作伙伴公司的专家可以对 File 进行修改，并且当有人恶意修改文件 File 时，能够动态解除他的修改权限。Alice 是 A 公司项目组 1 的成员，Bob 是 B 公司的专家，并且 Bob 在 C 公司有过修改文件 File 的历史。公司 A 的初始知识库如图 5.3 所示。

TIS1 表示 A 授予 Alice 属性“Member(Project1)”，TIS1 可以嵌入到证书当中颁发给 Alice，通过 TCP1 的计算规则，计算 Alice 的信任值为 1，用 SGP1 对 Alice 进行验证，由于 1>0.7，所以 Alice 获得角色 A.Administrator(File)继而获得修改 File 的权限。TIS2 表示 A 对 B 推荐 Expert 的信任值为 1，TIS3 表示 A 对 C 推荐 Modify(File)的信任值为 0.8，TIS2、TIS3 的初始值可由 A 灵活设定，可直接赋值，也可为其编写策略。

TIS1. A.Member(Project1) $\leftarrow$ Alice
TIS2. A.Expertrec(;1.0) $\leftarrow$ B
TIS3. A.Modifyrec(File;0.8) $\leftarrow$ C
SGP1. A.Administrator(File) $\leftarrow$ A.Administrator(File;?value > 0.7)
SGP2. A.Modify(File) $\leftarrow$ A.Administrator(File)
TCP1. A.Administrator(File;?value) $\Leftarrow$ 6*A.Member(Project1)
$\oplus$ 5* A.Expert(;?value) $\oplus$ 2*A.Modify(File;?value)
CRS1. A.Administrator(File;1) $\leftarrow$ Alice

图 5.3 公司 A 的初始知识库

当 A 与其他公司交流之后，A 从 B 公司和 C 公司处得到有价值的信任信息，并更新自身的知识库，更新后知识库 TIS 部分和 CRS 部分发生变化，发生变化的部分如图 5.4 所示。

图 5.4 显示知识库中增加了 6 条信任信息，TIS4 是从 B 收集的信息，表示 B 授予 Bob 专家属性，并附加信任值 0.8(B 可能依据与专家这个属性相关的其他属性，比如年龄，和相关的行为，比如在科研方面有过多少成功经验，对专家属性给出可信度)，TIS5 是从 C 收集的信息，表示 C 对 Bob 修改 File 的信任值为 0.9。TIS6、TIS7、TIS8、TIS9 是通过推演规则自动推出的结果。通过计算规则对 Bob 进行计算得 CRS2，因为 0.75>0.7，所以 Bob 也可获得修改 File 的权限。

TIS4. B.Expert(;0.8) $\leftarrow$ Bob
TIS5. C.Modify(File;0.9) $\leftarrow$ Bob
TIS6. A.Expert(;0.8) $\overset{rec}{\leftarrow}$ Bob
TIS7. A.Modify(;0.72) $\overset{rec}{\leftarrow}$ Bob
TIS8. A.Expert(;0.8) $\leftarrow$ Bob
TIS9. A.Modify(File;0.72) $\leftarrow$ Bob
CRS1. A.Administrator(File;1) $\leftarrow$ Alice
CRS2. A.Administrator(File;0.78) $\leftarrow$ Bob

图 5.4 信息交换后公司 A 的知识库

当 Alice 和 Bob 发生修改 File 行为时系统对其行为进行记录，并且根据记录进行计算，我们将时间片长度划分为 2^i 小时（i=0,1,2…），权值赋为 $1/2^i$，为经验和推荐分别赋权值 0.6、0.4，我们随机产生 Alice 和 Bob 的行为表现序列，在 Alice 和 Bob 发出修改 File 请求的时候更新知识库，更新的知识库部分如图 5.5 所示。

TIS9.A.Modify(File;0.77) ← Bob
TIS10.A.Modify(File;-0.4) $\overset{exp}{\leftarrow}$ Alice
TIS11.A.Modify(File;0.8) $\overset{exp}{\leftarrow}$ Bob
TIS12.A.Modify(File;0.9) ← Alice
CRS1.A.Administrator(File;0.65) ← Alice
CRS2.A.Administrator(File;0.79) ← Bob

图 5.5 信任值更新后公司 A 的知识库

CRS1、CRS2 表明用户的良好行为会增加信任值，恶意行为会降低信任值，Alice 因为信任值下降到阀值 0.7 以下，所以失去了角色 A.Administrator(File)，将不能够对 File 进行修改操作。

5.5 小结

本文介绍了一个基于角色策略和信誉的混合信任管理系统 RTE。RTE 信任策略语言通过在基于角色的信任关系语言中增加信誉值参数，从而能够在信任策略语言中支持信誉的管理。RTE 支持信任值的网络传递与计算，能够根据信任值进行细粒度的角色管理，并可根据角色行为的跟踪记录，动态管理角色授权，抗击角色域内的恶意行为。文章描述了 RTE 系统中策略语言的语法及推演规则，给出了信任值计算方法，并介绍了一个利用 RTE 系统进行资源访问控制的实例。

在开放网络环境中，数据的存储有时会以异地存储的方式存在，用户不知道自己的数据将会存储在什么地方，也不知道会有什么人可以看到自己的数据，数据的安全性将面临新的挑战，下一步的工作中，我们将基于 RTE 系统和 ABE 技术，将信任值作为属性，为用户分配密钥，实现数据的分布式安全存储。

第六章 面向复杂应用的信誉模型及工作流可靠性信任优化

为了能够整合多个资源来解决单个应用问题，许多 e-science 和 e-business 应用被组织成由多个作业顺序或并序组成的工作流。志愿计算（Volunteer Computing）是一种结合对等计算（Peer-to-Peer）和网格计算（Grid）的分布式计算技术，它能整合大量闲置资源进行复杂的科学计算，并被运用于多个应用如 SETI@Home 和 BOINC[123]。通常情况下，一个志愿计算环境包括大量地理上分布广泛的计算资源。这些资源的行为无法控制，且相互之间没有预先的信任。志愿计算的这种无信任计算环境给工作流任务的可靠性信任带来了诸多威胁，很多因素都会导致任务的失败，例如资源过载、网络连接慢、资源错误配置以及资源的恶意行为。因此在志愿计算环境中，一个工作流的调度除了考虑时间以外，还必须考虑工作流的可靠性。为此一个资源调度系统必须解决两个重要问题：(i)如何评估资源的可靠性，(ii)如何基于资源的可靠性信息进行面向可靠性的工作流调度。

为了评估资源的可靠性，很多分布式和多 agent 系统[8,86,91,109]采用信誉来跟踪评估资源的历史行为。然而目前的多数信誉模型存在两个问题：1.从资源的角度，大多数信誉模型[8,86,108,121]将资源的信誉定义为该资源成功完成作业数目的比率，他们忽略了作业运行时间（大小）对资源信誉的影响。例如在一个分布式系统中，资源 A 的作业失败率（单位时间内的作业错误数）比资源 B 高，因此资源 B 应该有比资源 A 更高的信誉。但是如果资源 A 运行的都是时间很短的作业，而 B 运行的都是时间很长的作业，传统的信誉模型将可能赋予 A 更高的信誉，因为资源 A 成功地完成了更多数目的作业。2.从作业的角度，大多数信誉模型将作业的成功率直接定义为该作业所在资源的信誉[87,121]，它们也没有考虑作业运行时间对作业可靠性的影响。例如在一个不可靠的资源上，某个作业运行的时间越长，它成功的概率应该越小。但在目前的信誉模型中，运行在某个资源之上的所有作业具有相同的可靠性。

基于资源的可靠性信息，为工作流任务进行资源调度以优化它的运行时间和可靠性是一个 NP-hard 问题[134]。为了能够给出面向时间[126,127,133]或面向可靠性优化的[121,135,136,138]工作流调度方案，目前的大多数研究工作采用基于列表的启发式技术进行资源调度。然而很多研究工作表明遗传调度算法(GAs)能够给出比基于列表的启发式更高质量的调度方案[87,127]。尽管遗传算法的时间复杂度更高，但这能够被运行时间较长的工作流任务接受。而且还可以采用并行遗传算法技术来克服该缺点[124]。

目前，由于遗传算法很难在随机演化调度方案的同时满足工作流作业间的相

关性，很少有遗传算法能够同时优化工作流的运行时间和可靠性。BGA (Bi-objective Genetic Algorithm) [137]是我们所知唯一关于该问题的遗传算法，但是 BGA 可能给出违反工作流作业相关性的无效调度方案。此外，现有的大多数遗传算法[128,137,143]采用随机演化机制，这可能导致算法的慢收敛。事实上，遗传算法的演化机制可以结合启发式技术从而能够更快地给出更好的结果，但目前很少有研究工作为遗传调度算法提出启发式规则。例如一些两阶段启发式规则被证明为性能最好的启发式[127]，但它们都无法在遗传算法中工作。

针对上文提出的信誉和调度问题，本章提出一种可靠性驱动 RD (Reliability-Driven)的资源信誉模型，并在此基础上设计一种前瞻的工作流遗传调度算法。RD 信誉模型采用与时间相关的作业失败率来定义资源的信誉，从而在信誉模型中考虑作业运行时间的影响。此外，文章给出的 RD 信誉算法能够实时地跟踪信誉的变化，并可被用来直接评估作业的可靠性。基于 RD 信誉，提出了前瞻的遗传调度算法 LAGA (Look-Ahead Genetic Algorithm)，它可对工作流的时间和可靠性同时进行智能的优化。LAGA 具有两个重要特点：1. 基于本章提出的资源优先级启发式，LAGA 的遗传算子可智能地变异调度方案，避免传统的随机变异带来的问题；2. 使用一种新颖的演化和评估机制：遗传算子（交叉和变异）只负责演化调度方案的作业资源映射，而调度方案的作业顺序由算法的评估步骤采用本章提出的 max-min 策略进行智能决策。本章提出的 max-min 策略是第一个能在遗传算法中运行的两阶段工作流启发式。依赖该策略，LAGA 能够避免 BGA 算法中的无效调度方案问题[137]，且能够通过智能的演化方法达到更好的收敛性。

本章的其它部分结构如下：第 2 节介绍相关工作，第 3 节描述志愿计算模型，第 4 节提出可靠性驱动的信誉管理，第 5 节定义基于 RD 信誉的工作流调度问题，第 6 节描述 LAGA 遗传调度算法，RD 信誉及 LAGA 算法的实验评估在第 7 节，最后一节总结全文。

6.1 相关工作

信誉系统可以实时监控分布式环境中资源的可靠性，并计算出资源能够成功提供某服务的期望概率[8]。在对等计算环境中，学者们提出了两种典型信誉系统 EigenTrust [108]和 PowerTrust [86]。他们将资源的本地信誉值定义为归一化的作业成功交互数。在志愿计算环境中，Sonnek 等人[121]将一个服务资源的可靠性定义为该资源正确作业响应的比率。在多 agent 系统中，Wang 和 Singh 等人 [90] 将信誉定义成一个三维的信念元组(b, d, u)，其中三个参数分别表示出现正确结果、错误结果以及不确定结果的概率。在贝叶斯信誉模型中，系统统计资源成功和失败作业

数的衰退总和，并由此计算资源的信誉。以上四种信誉模型都只考虑作业的数目，没有考虑作业运行时间的影响。Song 等人[87]采用模糊逻辑来评估信誉，虽然她们的信誉模型将作业的运行时间作为一个评估属性，但她们没有说明作业运行时间如何影响信誉的计算。此外，上述的大多数信誉模型都无法对服务资源的实时可靠性进行跟踪，而这却是作业调度系统所必需的。本文提出的 RD 信誉采用时间相关的作业失败率来定义信誉模型，RD 信誉算法可实时跟踪资源信誉的变化，进而可被直接用于作业可靠性评估。

在志愿计算环境中，一个工作流调度方案应同时优化任务的运行时间和可靠性。由于工作流的调度问题是一个 NP 问题[134]，很多调度算法采用基于列表的启发式[121,126,127,133,135]技术。为了优化工作流的时间，异构调度算法 HAS (Heterogeneous Scheduling Algorithm) [136]将一个作业的优先级定义为经过该作业的最长作业路径长度，并根据该作业优先级对各作业按次序进行调度。为了优化作业流的可靠性，Dongarra 等人[135]通过理论证明，作业调度应将尽量多的作业分配给具有最快指令执行速度和最佳可靠性的计算资源。Dogan 等人[138]提出一种双目标启发式规则 RDLS，它根据作业大小、作业开始时间、资源的计算能力和可靠性开销评估一个作业资源映射的优先级。文献[127]分析比较了一些工作流启发式的性能，指出两阶段启发式 min-min 具有最好的测试性能。由于 min-min 启发式规则需动态决定作业资源的映射，它不能在遗传算法中使用。本文在定义两个作业优先级启发式的基础上，提出一个两阶段 max-min 工作流调度策略，该策略能够结合遗传算法给出更好的作业调度方案。

通常情况下，遗传算法能够给出比基于列表的启发式更好的调度方案[127]。目前，BGA [137]是我们所知的唯一能同时优化工作流运行时间和可靠性的遗传算法。但是它可能给出违反工作流作业相关性的无效调度方案。为了在遗传算法随机演化的过程中保持作业的相关性，CorreÃ 等人[144]将工作流的作业划分为两个集合 V_1 和 V_2，使得 V_1 中的作业和 V_2 中的作业没有相关性。他们只对集合 V_2 中的作业顺序进行演化，从而避免了非法作业顺序的产生。Wang 等人[143]把工作流的调度方案表示成作业资源映射编码和作业顺序编码的组合，并对两种编码分别进行演化以避免无效调度方案的产生。尽管这两种方法能够解决无效调度方案问题，但它们都没有考虑工作流任务的可靠性。另外，大多数现有的遗传算法[128,137,143]采用随机演化的机制，会导致算法的慢收敛。本文提出的 LAGA 算法采用我们设计的 max-min 策略来决定调度方案的作业顺序，从而能避免无效调度方案的产生。另外，LAGA 采用的新颖演化评估机制能智能地加速调度方案的演化。

6.2 系统模型和假设

如图 6.1 所示，在一个典型的志愿计算模型中[121]，有一个中央服务器负责将客户提交的任务请求分配给系统中的资源。我们把一个工作流任务模型成一个有向无环图(DAG)：$Job=(V,E)$。V 是节点 $v_i(1\le i\le n)$ 的集合，每个节点表示工作流任务的一个子作业。E 是边 $e(i,j)(1\le i<j\le n)$ 的集合，$e(i,j)$ 表示作业 v_i 和 v_j 之间的相关性，其中 v_i 是父作业，v_j 是子作业。一个没有父作业的作业称为入口作业，一个没有子作业的作业称为出口作业。对于每一个作业节点 v_i，其权重 $|v_i|$ 是该作业必须执行的指令数目，这可以通过编译技术获得[135]。与其它一些工作类似[121,128,133]，本文只关注计算密集的工作流任务。我们没有在模型中考虑作业之间的通信延时，扩展我们的模型来包括通信时间将是我们下一步的工作。

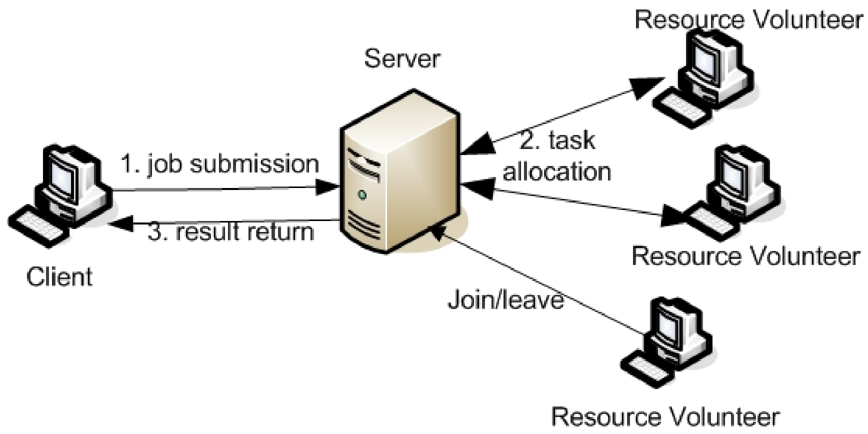

图 6.1 系统模型

一个志愿计算环境将包括很多资源志愿者，它们拥有不同的计算能力，且可以不受中央服务器控制地，自由加入或离开系统。假设集合 $R=\{r_1,r_2\cdots r_m\}$ 是系统当前可用的 m 个资源。每个资源 r_i 包括两个参数：资源的计算速度 γ_i 和资源的 RD 信誉 rdr_i。计算速度 γ_i 表示的是资源执行单个指令所需的时间。RD 信誉 rdr_i 描述资源的作业失败率，这会在下一节中介绍。基于系统资源的信息，中央服务器可以为每一个提交的工作流任务进行资源调度。假设 $M:V\to R$ 表示作业资源映射函数，则 $M(i)=r_j$ 表示作业 v_i 被分配给资源 r_j。工作流的调度问题和调度算法将分别在本章第 4 节和第 5 节中介绍。

本文中的其它假设条件还包括：假设中央服务器在任何时候最多只能分配一个作业给一个资源。另外我们也假设中央服务器能够监控作业的执行，即服务器能够检测到一个作业 v_i 的成功执行或者失败，并由此产生一个关于资源 $M(i)$ 的信誉报告。该假设的提出已得到多个相关研究工作的支持，如检查点技术和 quizzes

验证技术[141]都可以用来监测作业的执行。

6.3 可靠性驱动的信誉管理

在志愿计算环境中，由于资源的异构和不受控性，很多离散事件都会导致应用任务的失败，如服务失败、资源过载和计算资源的恶意行为。一般情况下，导致任务失败的事件都满足独立性和随机性，因此我们采用泊松分布来对资源的失效事件分布进行建模。失效事件的密度函数可表示为 $f(t)=\lambda e^{-\lambda t}(t\ge 0)$，其中 λ 是资源的失败率。假设在一个时间为 run_time 的作业运行期间内，某资源发生 num_fails 次作业失败事件，我们可以通过公式 6.1 来计算该资源的作业失败率，它是资源平均失效时间间隔(MTTF)的倒数。

$$\lambda=\frac{1}{\int_0^{\infty}\lambda x e^{-\lambda x}dx}=\frac{1}{MTTF}=\frac{num\_fails}{run\_time} \tag{6.1}$$

为了给一个工作流提供面向可靠性的资源调度，系统应监控资源的实时作业失败率。尽管传统的信誉模型能够监控资源的可靠性，但它们没有考虑作业时间的影响，且不能得到资源的作业失败率。而我们的可靠性驱动信誉模型直接和资源的失败率相关，可定义如下：

定义 6.1 一个资源 r_i 的可靠性驱动(RD)信誉(rdr_i)是该资源被广泛认可或相信的单位时间内作业失败的概率，即单位时间内该资源可能无法成功完成的作业平均数。

6.3.1 实时 RD 信誉计算

资源的 RD 信誉代表了资源的实时作业失败率 λ。为了计算 RD 信誉，我们把时间划分为连续的时间周期，每个时间周期持续一个窗口时间 T_{window}。对于每个时间周期，服务器为每个资源 r_i 维护一个信誉参数统计 $repu\_sta_i=(s_i,f_i,runtime_i,c_i)$。参数 s_i 和 f_i 分别表示一个时间周期的开始和结束时刻，$runtime_i$ 是资源 r_i 在本时间周期中用于作业运行的总 CPU 时间，c_i 是资源在本周期中遇到的作业失败数。如算法 6.1 所示，RD 信誉算法首先为每个资源 r_i 初始化第一个时间周期的信誉参数统计 $repu\_sta_i$（行 1~6），t_i 是资源 r_i 的当前时间周期编号。

Algorithm 6.1 RD Reputation Calculation Algorithm

1 for each resource r_i do
2 $rdr_i = rdr_i^0 = rdr^{initial}$
3 $t_i \leftarrow 1$
4 $s_i = f_i = current\_time$
5 $runtime_i \leftarrow 0;\ \ c_i \leftarrow 0$
6 end for
7 while there is a reputation record $testimony_j^i$ do
8 if ($f_j^i < s_i + T_{windows}$) then //current interval
9 $c_i \leftarrow c_i + c_j^i$
10 $runtime_i \leftarrow runtime_i + (f_j^i - s_j^i)$
11 $f_i \leftarrow \max(f_j^i, f_i)$
12 Remove the record $testimony_j^i$
13 Compute $\lambda_i^{statistic}$ by Equation 2
14 Compute rdr_i by Equation 3
15 else //next interval
16 $rdr_i^{t_i} \leftarrow rdr_i$
17 $t_i \leftarrow t_i + 1$
18 $s_i = f_i = s_i + T_{windows}$
19 $runtime_i \leftarrow 0;\ \ c_i \leftarrow 0$
20 end if
21 end while

在一个分配给资源 r_i 的作业 v_j 成功完成或者失败之后，服务器将产生一个信誉报告 $testimony_j^i = (s_j^i, f_j^i, c_j^i)$，$s_j^i$ 和 f_j^i 分别是作业 v_j 开始和完成时刻，c_j^i 是该作业运行期间的失败次数。如果一个作业失败，我们简单地将 c_j^i 赋值为 1，否则为 0。服务器将依据该报告更新信誉统计 $repu\_sta_i$（行 9~11），并利用公式 6.1 计算资源 r_i 在当前周期中的实时作业失败率 $\lambda_i^{statistic}$。这里，当前时间周期的总长度是 $f_i - s_i$。在本周期中，资源被监控到用于作业运行的 CPU 时间是 $runtime_i$，并遇到 c_i 次作业失败事件；在当前周期的其它时间 $f_i - s_i - runtime_i$ 中（服务器没有监控消息），资源的作业失败率被假设为资源上一个时间周期 $t_i - 1$ 的 RD 信誉（作业失败率）。因此，在当前周期的资源非监控时间中，资源的作业失败数被假设为 $rdr_i^{t_i - 1}(f_i - s_i - runtime_i)$，其中 $rdr_i^{t_i - 1}$ 是资源 r_i 上一个时间周期 $t_i - 1$ 中的 RD 信誉纪录。根据公式 6.1，资源的实时作业失败率可被定义为：

$$\lambda_i^{statistic} = \frac{c_i + rdr_i^{t_i - 1}(f_i - s_i - runtime_i)}{f_i - s_i} \tag{6.2}$$

一个资源的实时信誉应该随着时间而衰减，因此资源 r_i 在当前时间周期中的实时 RD 信誉可定义如下：

$$rdr_i = \alpha \cdot rdr_i^{t_i - 1} + (1-\alpha)\lambda_i^{statistic}, (0 \le \alpha < 1) \tag{6.3}$$

其中 α 是信誉的衰减因子。如果 α 为 0，那么实时 RD 信誉就等于 $\lambda_i^{statistic}$，这意味着实时信誉完全由当前的作业失败率决定。

在当前时间周期 t_i 的结束时刻（行 15），服务器把实时 RD 信誉 rdr_i 记录为资源 r_i 在本周期的信誉 $rdr_i^{t_i}$（行 16），并开始统计资源下一个时间周期 t_i+1 的信誉参数（行 17~19）。对于初始时间周期，我们假设每个资源 r_i 的 RD 信誉 rdr_i^0 为 $rdr^{initial}$（行 2）。$rdr^{initial}$ 是所有资源的初始 RD 信誉，它应被设为一个相对较高的作业失败率，从而能够激励资源提供者提供高质量服务以改善其信誉。

6.4 可靠性驱动的调度问题

基于实时 RD 信誉，我们可以形式化地定义可靠性驱动的工作流调度问题。在一个工作流任务中，每个作业必须在其所有父作业完成后才能被执行。因此，一个作业 v_i 可能的最早开始时间是：

$$t_i^{avail} = \max_{e(j,i)\in E} t_j^e \tag{6.4}$$

其中 t_j^e 是作业 v_j 的结束时间。如果作业 v_i 没有父作业，那么它的最早开始时间为 0。假设函数 $idle(r_j)$ 表示资源 r_j 能够空闲的时刻，则作业 v_i 的开始和结束时刻可分别定义为：

$$\begin{aligned} t_i^b &= \max\{t_i^{avail}, idle(M(i))\} \\ t_i^e &= t_i^b + |v_i|\gamma_j \quad where\ M(i) = r_j \end{aligned} \tag{6.5}$$

其中 $M(i)$ 表示作业 v_i 所在的资源，γ_j 是资源 r_j 的指令执行速度。假设 t_s^j 为工作流调度 S 中，资源 r_j 能够完成所有分配给它的作业的时刻，它可被定义如下：

$$t_s^j = \max_{i|M(i)=r_j}\{t_i^e\} \tag{6.6}$$

一个工作流任务的可靠性是其所有作业都能成功完成的概率，这可通过计算所有资源能够正常运行直至它被分配的作业都成功完成的概率来确定[135]。既然

RD 信誉 rdr_i 表示资源 r_i 的作业失败率，调度 S 中资源 r_i 能够成功完成其所有作业的概率可被计算为 $R_s^i = e^{-t_s^i \cdot rdr_i}$ 。因此如公式 6.7 所示，调度 S 中工作流任务的成功概率 R_s 可以计算为所有 R_s^i 乘积的结果。我们可以发现为了最大化任务的可靠性，系统需要最小化作业失败系数 $fal(S) = \sum_{i=1}^{m} t_s^i rdr_i$ 。

$$R_s = \prod_{i=1}^{m} R_s^i = e^{-\sum_{i=1}^{m} t_s^i \cdot rdr_i} \tag{6.7}$$

可靠性驱动的工作流调度应该在满足任务时间限制 D 的情况下，最大化任务的可靠性并最小化任务的运行时间。因此，可靠性驱动的工作流调度问题可以形式化为：

$$\begin{aligned} &\text{Minimize } fal(S) = (\sum_{i=1}^{m} t_s^i \cdot rdr_i) \\ &\text{Minimize } time(S) = \max_{r_i \in R} (t_s^i) \\ &\text{Subject to } time(S) < D \end{aligned} \tag{6.8}$$

6.5 可靠性驱动的工作流遗传调度算法

对于一个工作流任务，遗传算法通常能给出比基于列表启发式更好的调度方案。一个典型的遗传算法由以下几个步骤组成：(1)创建一个由随机算法产生的调度方案（染色体）初始群体；(2)评估当前群体中每个调度方案的适应度，并为下一代群体选择调度方案；(3)运用两种遗传算子（交叉和变异）生成新一代群体的调度方案；(4)重复步骤 2 和 3 直到调度方案群体收敛。通常情况下，遗传算子对调度方案采用随机演化的方法[128,137,143]，这会产生违反作业相关性的无效调度方案，也会导致算法的慢收敛。为了解决这个问题，本文设计了一个前瞻的遗传调度算法 LAGA（Look-Ahead Genetic Algorithm）。LAGA 使用一种新颖的演化和评估机制，它没有像其它遗传调度算法一样在遗传算子中演化作业的执行顺序，而是在算法的评估步骤采用我们提出的 max-min 策略对作业顺序进行智能调度。下面将对 LAGA 算法的各个部分分别进行介绍。

A．染色体编码

在遗传调度算法中，一个工作流任务的调度方案将被编码为一个数据结构，也称为一条染色体。如图 6.2c 所示，本文采用一个二维编码串来表示一个调度方案。编码串的第一个维表示资源的索引，它描述了作业资源映射；编码串的另一

个维描述作业间的调度顺序。在算法 LAGA 中，遗传算子只负责演化调度方案的第一维作业资源映射，这可通过将调度方案的二维编码转换成作业资源映射编码串 M 来实现。如图 6.2d 所示，编码串 M 是一个长度为$|V|$的向量，它与作业资源映射函数具有相同的含义，即 $M(i)=r_j$ 表示作业 v_i 被分配给资源 r_j。

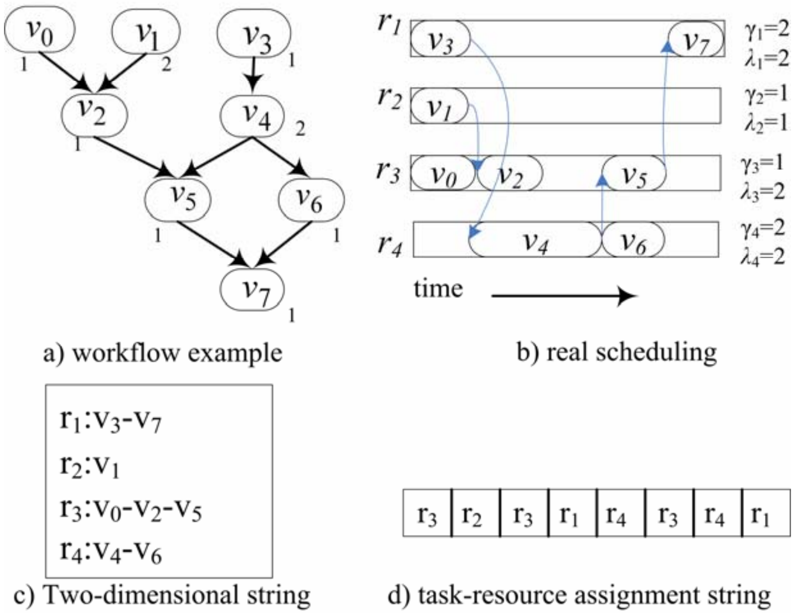

图 6.2 编码例子

B．交叉遗传算子

在遗传算法中，交叉遗传算子希望能通过交换两条适应度很好的染色体以便能产生一条适应度更好的染色体。为了保持工作流作业间的相关性，一些研究工作设计两个交叉算子分别对调度方案的作业资源映射和作业执行顺序进行随机交换[128,143]。这种方法将会导致交叉算子很难找到一个适应度很高的调度方案，因为对某个作业资源映射而言是适应度高的作业执行顺序，并不一定对另一个作业资源映射也是适应度高的。我们认为在调度方案的作业资源映射确定后，它的作业顺序不应该使用随机交换的方法来确定，可以利用一些启发式规则进行智能优化。因此在我们的 LAGA 中，交叉遗传算子仅对调度方案的作业资源映射进行演变，而调度方案的作业顺序将在后面的算法评估步骤中采用 max-min 两阶段调度策略来决定。

在 LAGA 算法中，交叉遗传算子首先随机地以概率 p_c 从当前调度方案群体中选择一些染色体对。对于每一对染色体，算法在它们的作业资源映射编码串 M 中随机产生一个切断点，把这一对编码串划分成顶端和末端两部分。然后，算法交换两个编码串的末端部分，从而产生两个新的作业资源映射编码串。交叉遗传算子的操作过程如图 6.3 所示。

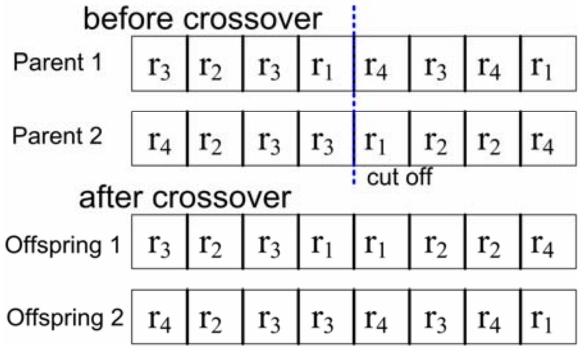

图 6.3 交叉遗传算子

C．变异遗传算子

遗传算法的变异算子可以将算法带离可能的局部搜索困境。一般情况下，变异遗传算子会随机改变一条染色体中的一些基因，这可能会导致算法在好的调度方案外围随意漫步的问题。为此，我们的变异算子对传统的操作进行改进，利用资源优先级启发式对调度方案进行智能地变异。为了优化工作流任务的可靠性，文献[135]通过理论证明得到结论：应将尽可能多的作业分配给指令速度（单个指令的执行时间）与作业失败率乘积最小的资源。因此我们可以定义如下的资源优先级启发式：

定义 6.2 资源优先级启发式(ResPH)：假设资源 r_i 的优先级是 $1/\gamma_i rdr_i$ ，S 是一个工作流调度，且满足 S 中所有的作业都被分配给具有最高优先级的资源。那么，任何 $S' \neq S$ ，R_S 和 $R_{S'}$ 分别为调度 S 和 S'中工作流任务的可靠性，都有 $R_{S'} < R_S$ 。

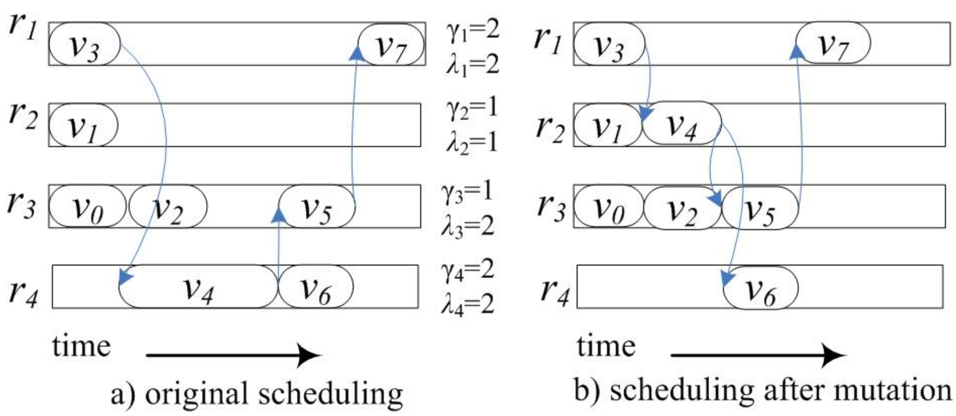

图 6.4 变异遗传算子

和交叉遗传算子相似，LAGA 的变异算子仅对调度方案的作业资源映射进行演化。它首先以概率 p_m 在现有调度群体中随机选择一个调度方案，然后再在该调度方案中随机选择一个作业，并将该作业重新分配给一个具有更低 $\gamma_i rdr_i$ 乘积的资

源。如图 6.4a 所示，某调度方案开始将作业 v_4 分配给 $\gamma_i rdr_i$ 乘积为 4 的资源 r_4，在变异算子操作后，作业 v_4 被重新分配给具有更低的 $\gamma_i rdr_i$ 乘积（$\gamma_i rdr_i = 1$）的资源 r_2。在图 6.4b 显示的新的调度方案中，该工作流任务的时间和可靠性都得到了改善。

D．评估

在评估步骤中，大多数遗传算法仅对调度方案的各种参数进行评估，它们不会根据参数的评估来改进调度方案的质量。由于 LAGA 的遗传算子只演化了调度的作业资源映射，因此算法的评估步骤将首先需要为每一个新的调度方案决定作业执行顺序，然后再评估每个作业的估计结束时间，从而能够利用公式 6.5 评估新调度方案 S 的运行时间 $time(S)$ 和失败系数 $fal(S)$。

为了能为一个特定的作业资源映射编码串给出一个优化的作业执行顺序，我们首先定义两个作业优先级启发式，然后基于这两个优先级启发式提出一种新的 max-min 两阶段作业顺序调度策略。为了优化一个工作流任务的运行时间，一些能尽早启动以及对任务运行时间有很大影响的作业应拥有更高的资源调度优先级。因此我们的第一个作业优先级启发式可定义为：

定义 6.3 作业优先级启发式 1(TaskPH1)：设某作业 v_i 对工作流任务的重要性为工作流 DAG 图中从该作业开始的最长路径的长度，它可表示如下：

$$impt(i) = \begin{cases} |v_i| & \text{if } v_i \text{ is an exit task} \\ |v_i| + \max\limits_{e(i,j)\in E} impt(j) & otherwise \end{cases} \tag{6.9}$$

设 $E(\gamma)$ 表示所有资源的平均指令速度，则作业 v_i 的优先级 $p(i)$ 可定义为：

$$p(i) = E(\gamma)\cdot impt(i) - \max(t_i^{avail}, idle(M(i))) \tag{6.10}$$

如果两个作业被分配到同一个资源，那么具有更高优先级的作业应得到优先执行。*TaskPH1* 利用资源的平均指令速度来评估从某作业开始的最长路径的完成时间，该方法的优点是容易实现。在我们的遗传算法中，由于前面的遗传算子已经确定了调度方案的作业资源映射，因此，我们可以更加准确地评估一条路径的完成时间。第二个作业优先级启发式可定义为：

定义 6.4 作业优先级启发式 2 (TaskPH2)：在工作流 DAG 图中，设从作业 v_i 开始的最长路径的估计完成时间是：

$$comp(i)=\begin{cases}|v_i|\cdot\gamma_{\mathrm{j}} & \text{if } v_i \text{ is an exit task}\\ |v_i|\cdot\gamma_{\mathrm{j}}+\max\limits_{e(i,k)\in E} comp(k) & otherwise\end{cases}$$
$$\text{where } M(i)=r_j \tag{6.11}$$

则作业 v_i 的优先级 $p(i)$ 为：

$$p(i)=comp(i)-\max(t_i^{avail},idle(M(i))) \tag{6.12}$$

基于上面提出的作业优先级启发式，LAGA 的评估算法将使用一个两阶段 max-min 策略对作业的执行顺序进行调度。如算法 6.2 所示，基于 *TaskPH1* 或者 *TaskPH2*，算法首先为每个资源选择一个具有最高优先级的作业作为下一个调度的作业。然后，算法从所有资源的下一个调度作业中，选择具有最小结束时刻的作业进行作业顺序调度。算法的输入为一个作业资源映射编码串 *M*，输出为分配到每个资源 r_i 上所有作业按调度顺序排列的队列 que_i，以及在新的调度方案 S 中的每个资源的作业估计完成时间 t_S^i。算法中，队列 que_ready_i 包含资源 r_i 上已准备好运行但尚未被调度的作业。

LAGA 的工作流程包括：(i)把每一个入口作业 v_j 添加到其所分配资源 $M(j)$ 的作业就绪队列中，并将该作业的有效开始时间设置为 0（行 3~6）；(ii)为每一个资源选择具有最高优先级的作业（行 11）；(iii)在所有被选择的作业中，选择具有最小结束时刻的作业 v_{task_sel}（行 12~16）；(iv)为最终选择的作业 v_{task_sel} 安排执行顺序（行 18~20），并更新资源 $M(task\_sel)$ 的作业完成时间和空闲时间（行 21）；(v)更新作业 v_{task_sel} 所有子作业的状态（行 22~25）；(vi)重复步骤 ii-v 直至所有作业的执行顺序都被确定。

定理 6.1 评估算法的时间复杂性为 $\mathrm{O}(n\log n+nm+d)$，其中 m 是资源的数目，n 是 DAG 工作流中的节点（作业）数目，d 是 DAG 中有向边（相关性约束）的数目。

证明. 评估算法中，初始化作业就绪队列的时间复杂性是 O(n)（行 3~6）。算法的一次完整循环（行 7~26）将会为一个作业的顺序进行调度，因此算法的循环将会运行 n 次。为了能够有效地对作业优先级进行排序，并为每个资源选择一个最高优先级作业（行 11），需要耗时 $\mathrm{O}(\log n)$。计算作业的结束时间并选择具有最小结束时间作业的时间复杂性是 O(m)（行 12~16）。算法行 18~21 的时间复杂性是 O(1)。因此 n 次算法循环（行 8~21）的时间复杂性是 $\mathrm{O}(n(\log n+m+1))$。为了更新子作业的状态（行 22~25），算法耗时 O(d)。所以，评估算法总的时间复杂

性是 $O(n+n(\log n+m+1)+d)=O(n\log n+nm+d)$。

Algorithm 6.2 The Evaluation Algorithm

input: task-resource mapping string M
output: $\{t_S^i, que_i\}$ for each resource r_i
for each entry task v_j
    add v_j to the task ready queue $que\_ready_{M(j)}$
    $t_j^{avail} \leftarrow 0$
end for

repeat
    $min\_end \leftarrow \infty$ //the minimum ending time
    $task\_sel \leftarrow null$ //the task selected
    for each resource r_i
        //max-min phase 1
        find the task v_j with the maximum priority value from que_ready_i
        //max-min phase 2
        compute the ending time t_j^e for v_j using equation 2
        if $t_j^e < min\_end$
        $min\_end \leftarrow t_j^e$ //the minimum ending time
        $task\_sel \leftarrow j$ //record the selected task
        end if
    end for

    $res\_sel \leftarrow M(task\_sel)$
    remove v_{task_sel} from $que\_ready_{res\_sel}$
    add v_{task_sel} to que_{res_sel}
    $t_S^{res\_sel} = idle(res\_sel) = t_{task\_sel}^e$
    for each child task v_i of task v_{task_sel}
        update t_i^{avail} using equation 1
        if v_i is ready to run, add it to $que\_ready_{M(i)}$
    end for
until every que_ready_i is empty

E．选择

在遗传算法中，适应度函数被用来测量和选择调度方案。由于算法的目标是在时间约束下同时优化一个工作流任务的运行时间和可靠性，因此我们采用 SWGR (sum of weighted global ratios) 模型[137]来计算调度方案 S 的适应度，它可以

被定义为：

$$f(S) = \omega_1 \cdot \frac{fal(S)-\mathrm{minFal}}{\mathrm{maxFal}-\mathrm{minFal}} + \omega_2 \cdot \frac{time(S)-\mathrm{minTime}}{\mathrm{maxTime}-\mathrm{minTime}}$$

$$+ penalty(S) \qquad (\omega_1+\omega_2=1)$$

$$where \quad penalty(S) = \begin{cases} 0 & if \quad time(S)<D \\ 1 & if \quad time(S)>D \end{cases} \tag{6.13}$$

其中maxFal和minFal分别表示现有调度群体中调度方案的最大失败系数和最小失败系数，而 maxTime 和 minTime 分别表示调度方案的最大时间和最小时间。函数 $f(S)$ 的前两项鼓励算法选择具有最小失败系数和最小运行时间的调度方案，这两个目标可以根据用户的需求设定不同的权重。第三项 $penalty(S)$ 处理算法的时间约束目标，如果调度方案的时间超过了时间约束 D，它将会受到一个适应度惩罚。为了产生下一代调度方案群体，现有的所有调度方案将首先按照其适应度 $f(S)$ 进行降序排序，然后算法采用轮盘赌选择法（roulette wheel selection scheme）为下一代群体选择调度方案。轮盘赌选择法[143]被遗传算法广泛使用，本文由于篇幅问题就不重复介绍了。

6.6 实验和结果

为了验证 RD 信誉以及 LAGA 算法的性能，本节将首先介绍我们模拟的志愿计算环境。然后对比分析 RD 信誉对信誉计算以及资源调度的影响。基于 RD 信誉，我们将比较 LAGA 与两个典型的基于列表的启发式以及另外一个遗传算法的性能，同时，我们也将评估 LAGA 算法中的三种优先级启发式的效果。

6.6.1 实验环境

本文使用 GridSim[125]将志愿计算环境模拟为三个参数：资源的数目 m，资源的平均速度 γ 和资源的平均作业失败率 λ。参照一些其他相关工作[126,136,137]的参数设置，本文的志愿计算环境包括 200 个资源志愿者，它们提供不同的 CPU 计算能力，其速度均匀分布在 5×10^{-4} 和 10^{-3} 之间（单位：毫秒/指令）。资源志愿者的作业失败率均匀分布在 $10^{-3}/h$ 到 $10^{-4}/h$ 之间[137]。和其它很多工作一样[126,137,143,145]，本文采用随机 DAG 图生成器将一个工作流任务模拟为三个参数：子作业数目，作业节点的平均出度以及作业的平均大小。实验中，一个工作流任务的作业数目在 40 到 200 之间，作业节点的平均出度为 2，作业的大小均匀分布在 1×10^4 MI (Million instructions)和15×10^6 MI之间。

对于系统中的其它参数，所有资源的初始信誉$rdr^{initial}$设为$10^{-3}/h$，信誉衰减系数α为0.2。适应度评估函数中两个权重因子ω_1和ω_2都设为0.5，这意味着算法中任务的可靠性和运行时间具有相同的优先级。交叉算子的选择概率p_c为0.5，变异算子的选择概率p_m为 0.25。这两个概率参数都被设为文献[124]使用参数的中间值，这样使我们能够测试遗传算法的演化过程。LAGA 算法的调度方案群体大小为 20。实验中，对于具有相同参数的每种工作流任务，我们创建 5 个实例以便能获得更具代表性的工作流评估数据。另外，对于每个工作流任务实例，我们将运行遗传算法 3 次以获得遗传算法的平均性能。

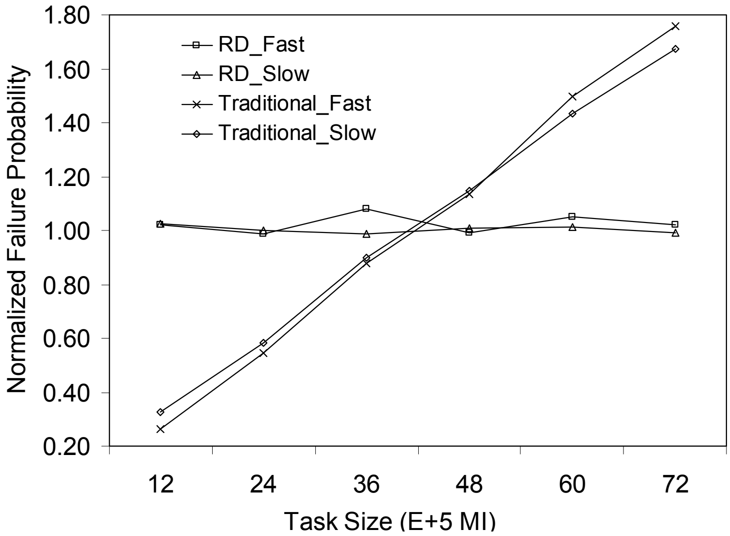

图 6.5 不同资源速度情况下作业的归一化的失败概率

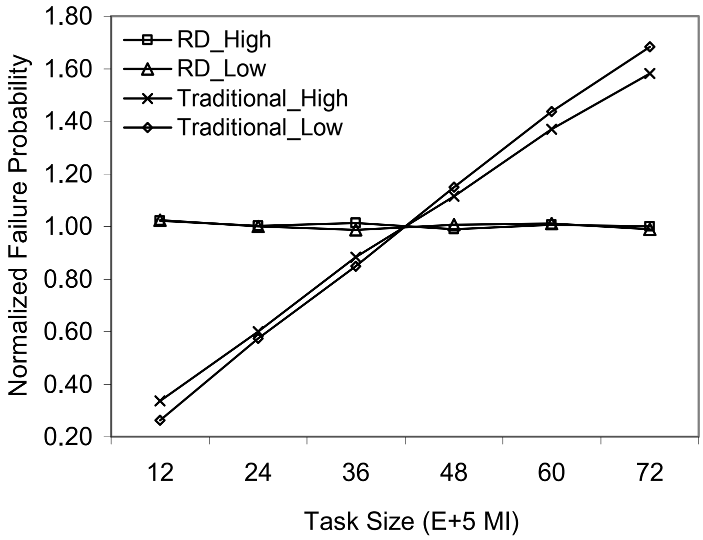

图 6.6 不同的资源失败率情况下作业的归一化的失败概率

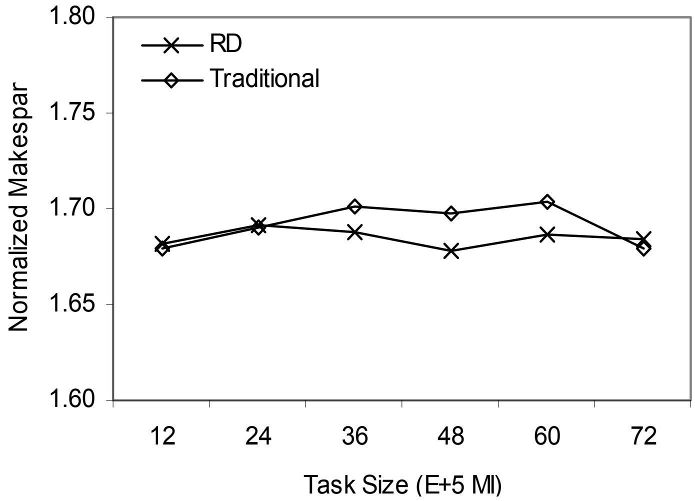

图 6.7 基于传统信誉和 RD 信誉的工作流时间

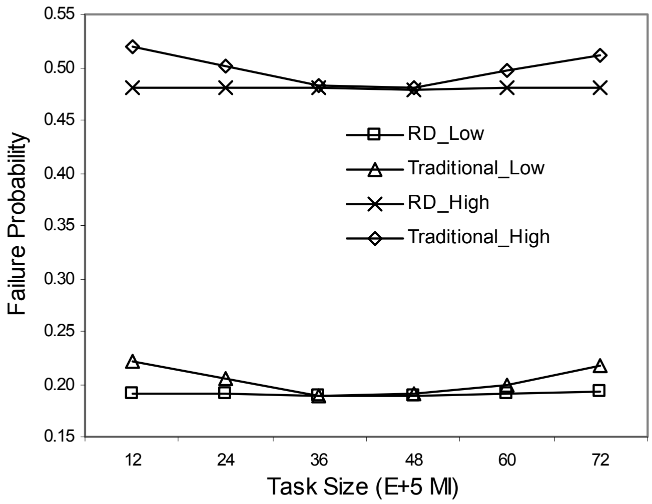

图 6.8 基于传统信誉和 RD 信誉的工作流失败概率

6.6.2 RD 信誉评估

a) RD 信誉和传统信誉比较

传统的信誉模型将资源的信誉定义为该资源成功完成作业数目的比率。为了比较 RD 信誉和传统信誉的不同，我们在以下几种条件下测试两种信誉的性能：系统中所有测试作业的大小分别是$\{12,24,36,48,60,72\}\times 10^5$ MI，资源志愿者的作业失败率分别为$10^{-3}/h$（高）或$10^{-4}/h$（低），资源的 CPU 速度（单指令执行时间）分别为 1000MIPS（快）或 500MIPS（慢）。由于 RD 信誉和传统信誉有着不同的单位，为了便于比较，我们将比较基于两种信誉得到的一个中等大小作业的作业失败概率。图 6.5 和图 6.6 显示的是归一化的两种作业失败概率，它可被计算为（根据信誉评估得到的作业失败概率 / 基于资源实际失败率得到的作业真实失败概

率）。由图 6.5 可以发现，基于 RD 信誉得到的作业失败概率与真实作业失败概率几乎一致。但是基于传统信誉的作业失败概率仅仅在系统中的测试作业也具有中等作业大小时，才与真实的作业失败概率接近，否则作业失败概率将随着测试作业大小的增大而增长。另外，当志愿资源有着更快的速度（图 6.5）或者更低的失败率时（图 6.6），基于传统信誉的作业失败概率会偏离真实值更远。这是因为基于传统信誉的归一化作业失败概率是一个负指数函数。资源更低的失败率和更快的速度将会使得函数的指数更小，从而产生更大的偏差。

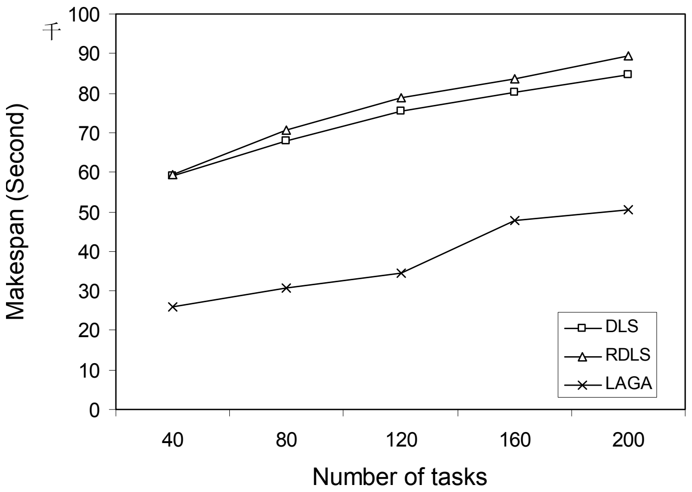

图 6.9 启发式 DLS, RDLS 和 LAGA 给出的调度方案运行时间

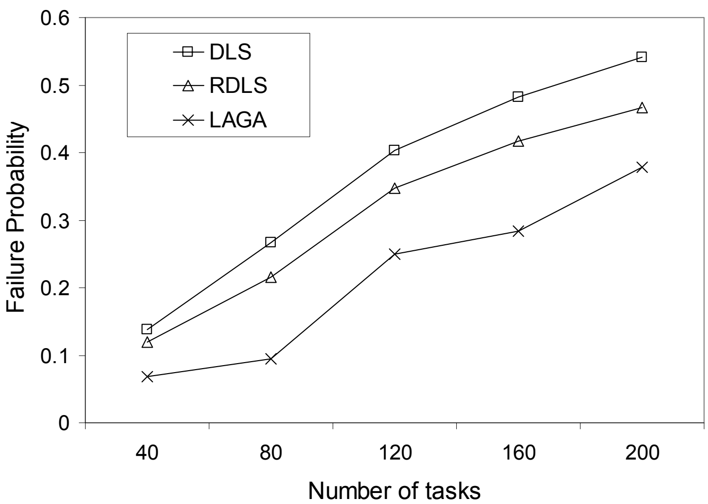

图 6.10 启发式 DLS, RDLS 和 LAGA 给出的调度方案作业失败率

b) RD 信誉对任务调度的影响

本小节将分析传统信誉和 RD 信誉对任务调度的影响。实验中系统的一半资源具有实际的作业失败率，另一半资源具有基于 RD 信誉的作业失败率或者基于传统信誉的作业失败概率。图 6.7 显示，在各种不同的测试条件下，基于传统信誉和基于 RD 信誉的调度方案几乎具有相同的任务时间。图 6.8 则显示基于 RD 信誉的调

度方案比基于传统信誉的调度方案具有更低的作业失败概率，尤其是在测试作业非常小或者非常大的两种情况下，基于 RD 信誉的调度方案可靠性更加突出。这是因为在这两种情况下，传统的信誉模型会给出一个和真实值不同的资源失败率，这将导致更多的作业被分配给可靠性更低的资源。

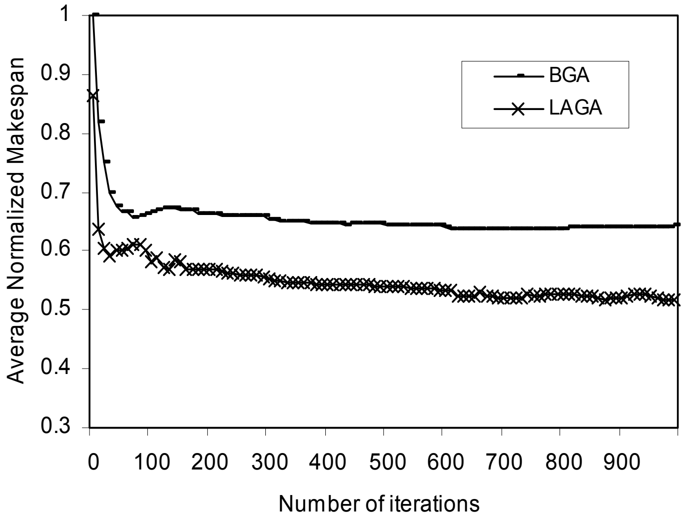

图 6.11 遗传算法每次循环的调度方案平均归一化运行时间

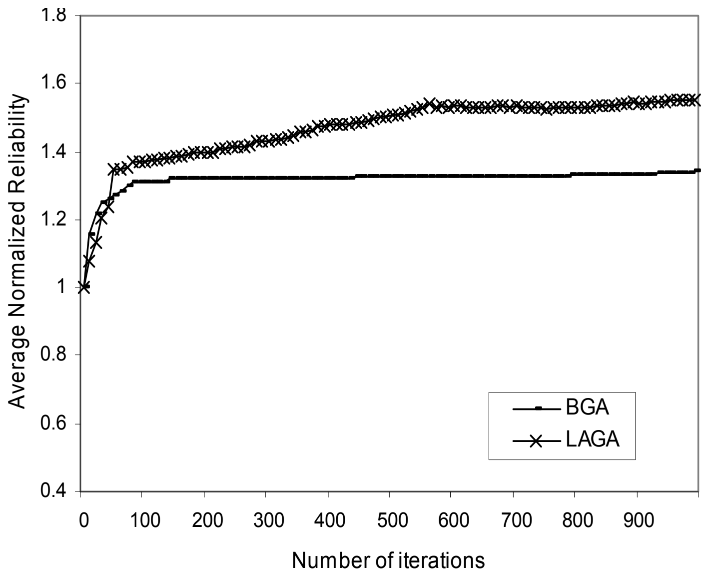

图 6.12 遗传算法每次循环的调度方案平均归一化可靠性

6.6.3 LAGA 算法性能

a) LAGA 和基于列表的启发式规则

基于列表的启发式规则 DLS 和 RDLS[138]能够分别对工作流任务的时间和可靠性进行优化，并得到广泛认可[126,136]。下面，本文将比较 LAGA 和这两种启发式规则的性能。被测试的工作流作业数目在 40 到 200 的范围内变化，对于每个工作流

实例，我们分别运行 DLS 和 RDLS 启发式 100 次以求得它们的平均结果。图 6.9 和图 6.10 显示，从任务的时间和可靠性两个角度，LAGA 都能为一个工作流任务提供最好的调度方案。尤其当工作流作业数目很少的时候（40 个作业），LAGA 的性能比其它两种启发式规则提高得更多（大约 15%）。这是因为当作业数目很少的时候，系统将会有更多的空闲资源供作业选择。因此，LAGA 将会对它们逐个检查从而选出最合适的资源。但是基于列表的启发式规则只会根据启发式值检查一个资源，而该资源很可能不是最合适的资源。

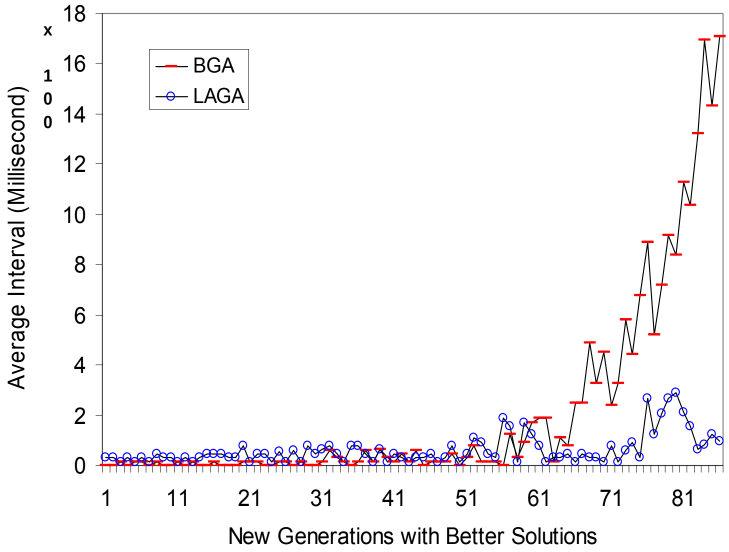

图 6.13 算法 BGA 和 LAGA 演化所需时间

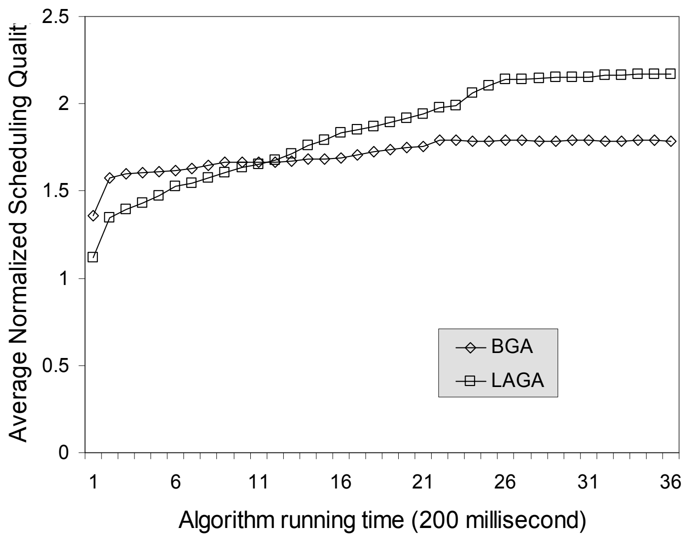

图 6.14 算法 BGA 和 LAGA 随时间的演化性能

b) LAGA 和另一个遗传算法

BGA [137]也是一个同时优化工作流任务时间和可靠性的遗传调度算法，它采取随机的方法对调度方案进行演化。下面，本节将从算法循环和算法时间两个角度

比较两种遗传算法的性能。遗传算法的每次循环都将产生新一代调度方案群体，为了比较算法的循环性能，本文统计各调度方案群体的平均归一化任务时间和归一化可靠性。归一化的任务时间（或可靠性）可定义为现有调度方案群体的平均时间（或失败系数）除以初始调度方案群体的平均时间（或失败系数）。被测试的工作流任务作业数目为 200，两个遗传算法的循环次数设为 1000。图 6.11 和图 6.12 显示 LAGA 能够比 BGA 更快地演化工作流任务的时间和可靠性。另外，在相同算法循环次数内，LAGA 总能给出比 BGA 更好的调度方案。

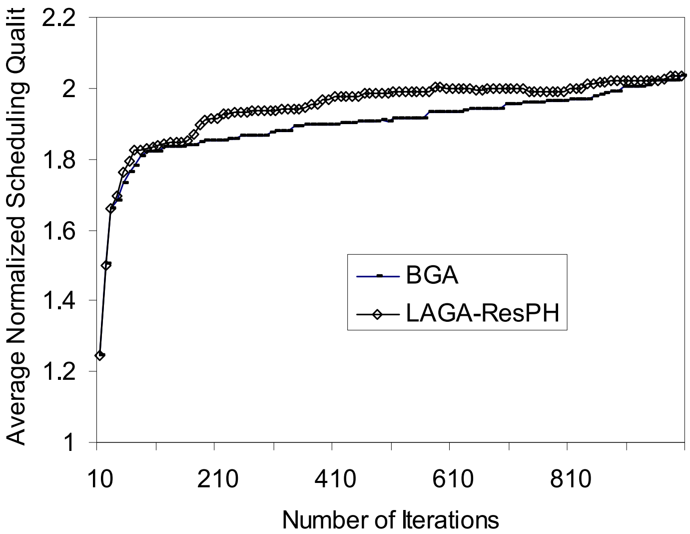

图 6.15 启发式 ResPH 的有效性

为了从时间角度比较算法的演化性能，我们将做两个实验。在第一个实验中，文章将统计算法产生新一代调度方案群体（含有更好质量的调度方案）所需的平均时间。图 6.13 显示，在算法演化的初期，BGA 产生新一代群体所需的时间比 LAGA 少（平均少 16 毫秒）；但在算法演化的末期（62 代以后），BGA 比 LAGA 需要更多时间来产生新一代群体。这是因为在算法演化的初期，BGA 很容易通过随机演化的方法找到一个更好的调度方案；但是随着演化过程的进展，随机演化的方法将越来越难找到一个更好的调度方案。与 BGA 相反，LAGA 采用一个时间复杂度较大的启发式演化方法，因此它在演化的初期比 BGA 慢；但是，LAGA 的启发式演化方法能够保证算法在演化的末期仍然能在相对较少的时间内找到更好的调度方案。

在第二个实验中，我们比较两个遗传算法的平均调度质量在算法运行时间维度上的演化进度。文章每隔 200 毫秒对两个算法的归一化调度质量进行采样。一个遗传算法的归一化调度质量是算法归一化可靠性以及归一化任务时间倒数的和。如图 6. 14 所示，在演化的初期，BGA 可以比 LAGA 更快地改进调度方案的质量；然后，BGA 改进调度方案的速度变得越来越慢；最终，LAGA 的性能超过 BGA，并能提供质量更好的调度方案。图 6.14 的结果也证明我们对第一个实验（图

6.13）的分析是正确的。

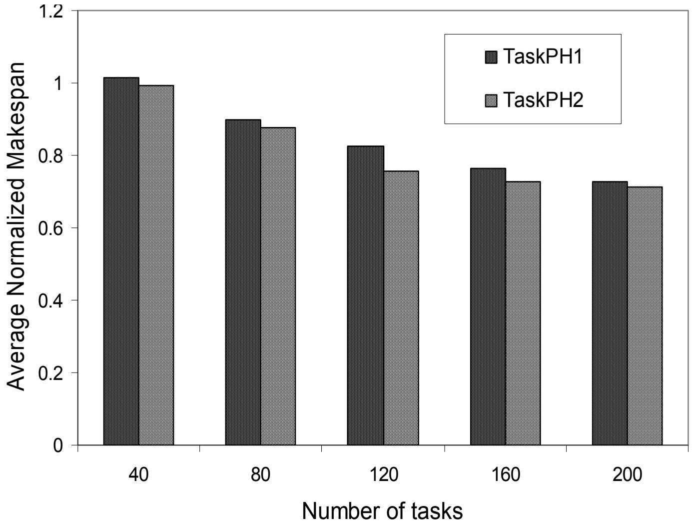

图 6.16 使用 TaskPH1 或 TaskPH2 给出的调度方案运行时间

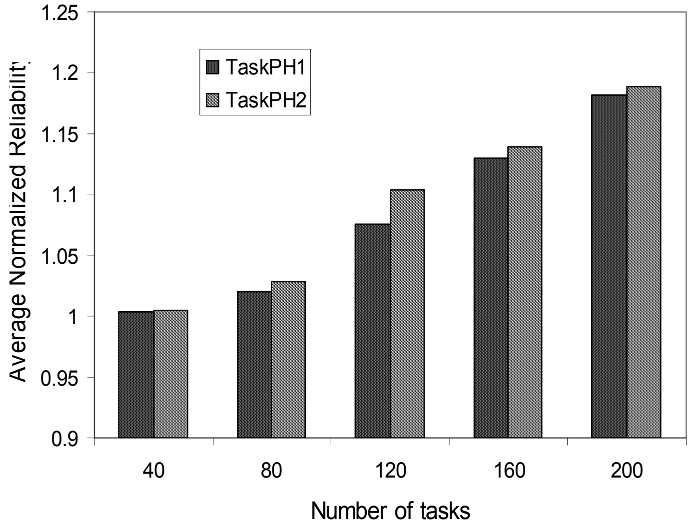

图 6.17 使用 TaskPH1 或 TaskPH2 给出的调度方案可靠性

6.6.4 优先级启发式的有效性

a) ResPH 的有效性

为了评估资源优先级启发式 ResPH 的有效性，我们实现一个仅使用 ResPH 启发式的 LAGA，并比较它与 BGA 的性能。被测试的工作流任务作业数目为 200。图 6.15 显示了两种遗传算法在每次算法循环时的平均归一化调度质量（和上面的定义相同）。可以看到，ResPH 在演化初期能够帮助遗传算法给出更好的调度方案；但在算法结束时，ResPH 不再能继续改进调度质量，两种遗传算法给出几乎相同质量的调度方案。这表示在遗传算法演化一段时间后，算法将很难通过把一

个作业分配给更快或更可靠的资源来提高一个调度方案的质量。

b) TaskPH1 和 TaskPH2 的有效性

本小节我们将比较作业优先级启发式 TaskPH1 和 TaskPH2 对算法 LAGA 的影响。实验中，工作流任务的作业数目在 40 到 200 之间。图 6.16 和图 6.17 给出了使用 TaskPH1 或 TaskPH2 的两种 LAGA 的性能，该性能指标包括：两种 LAGA 算法给出的调度方案的平均任务时间和可靠性分别除以 BGA 算法给出的调度方案的任务时间和可靠性。

我们可以发现 TaskPH1 和 TaskPH2 都能帮助 LAGA 给出比 BGA 更少时间（归一化时间< 1）和更高可靠性（归一化可靠性> 1）的调度方案。进一步分析可以发现，当工作流任务的作业数目较小（40 个作业）或较大时（200 个作业），TaskPH1 和 TaskPH2 对算法 LAGA 有着差不多的优化性能；但当工作流任务的作业数目为中等大小（120 个作业）时，TaskPH2 对遗传算法的优化能力比 TaskPH1 有了显著提高。这是因为当作业数目很小的时候，遗传算法即使没有启发式也能找到一个好的调度方案；但当作业数目很大时，每个资源都被分配了很多作业，这使得一条作业路径的完成时间很难被评估。因此在这两种情况下，虽然 TaskPH2 可以比 TaskPH1 更准确地预测一条作业路径的完成时间，但它的性能优势不能得到体现。

6.7 小结

本章研究了志愿计算环境中可靠性驱动的工作流调度问题。为了评估资源的可靠性，文章提出了时间相关的可靠性驱动 RD 信誉模型。RD 信誉使用资源的作业失败率来定义信誉，从而使调度系统可以直接使用 RD 信誉来评估一个作业的可靠性。另外文章设计的 RD 信誉算法能够实时监控资源信誉状态的变化。

基于 RD 信誉，本章提出了一个前瞻的遗传调度算法 LAGA 来智能地为工作流任务同时优化时间和可靠性。LAGA 采用我们提出的资源优先级启发式对传统遗传算法的变异算子进行优化，并使用一种新颖的演化和评估机制：遗传算子只负责演化调度方案的资源作业映射，而调度方案的作业顺序由算法的评估步骤采用我们提出的 max-min 策略进行智能决策。实验结果显示 RD 信誉能够计算出更准确的资源信誉，并能在工作流调度中增加一个工作流任务调度的可靠性。LAGA 算法能够给出比基于列表启发式 DLS 和 RDLS 更好的调度方案，而且，它比另一种遗传算法 BGA 有着更好的调度方案演化性能。

第七章 总结与展望

7.1 本文总结

在分布协同的互联网环境中，本文根据信任的需求层次将信任管理分为四个递进的层次：基于身份的信任、服务属性信任、面向可靠性的信任、面向健壮性的信任。针对前三个层次的信任管理，我们提出紧密相关的四个研究问题：如何设计出功能强大的基于身份的信任策略描述语言、如何得到一个全面且健壮的服务信誉模型、如何结合基于策略和信誉的信任管理以及如何使信任概念在面向可靠性的复杂网络系统中得到应用。为了解决这四个信任问题，本文介绍了我们的解决方案，下面分别对我们的工作进行总结。

作为跨安全域资源访问管理的主要方法之一，自动信任协商系统的策略语言需能支持复杂的访问控制和信任协商策略，并能高效的放置和搜集信任证书。本文提出一种面向信任分布式证明和协商的策略语言 RTP。RTP 对 RT 语言进行改进和拓展，具有很强的访问控制策略描述能力，可以定义复杂的角色；语言中增加 lsign 语法，可以定义逻辑推导角色，能够支持信任分布式证明；语言的信任协商策略增加 release 谓词，从而可以保护信任证书敏感信息的传播；信任协商策略中增加 prove 和 find 谓词，可以定义信任协商启发式规则，避免信任证书的盲目搜索。为了体现 RTP 语言的全面功能，本文还提出一个基于 RTP 语言的信任分布式证明协商算法 DPN。DPN 算法通过本地信任协商和远程信任证明，可以高效地完成信任分布式证明任务。文章详细介绍了 RTP 语言的语法构成，定义了 RTP 语言的推理证明规则，给出了语言的语义解释并证明了语言的可靠性和完全性。文章通过信任图的概念分析了算法的正确性和完整性。我们的实验表明，跟传统的信任协商方法相比，DPN 算法能够有效地减少信任建立时间和交互次数。

针对目前信誉模型不能评估信誉估计方差以及相加的反馈聚合方法难以支持健壮性信誉评估两个问题，本文描述了一个健壮的线性马尔科夫信誉模型 RLM。RLM 模型将信誉评估表示成信誉估计值和信誉估计方差两个属性，并采用线性自回归方程定义信誉状态空间的演化。由于 RLM 模型构成了一个隐马尔科夫过程，模型采用完全基于统计推测理论的卡尔曼滤波方法，它通过参数反馈噪音方差为健壮性的统计推测技术提供了支持。为了给出一个健壮的模型参数校准从而能抵抗恶意反馈攻击，模型首先采用 EM 参数动态校准算法，该算法可以自动给出合适的参数选择，从而能减轻不正确信誉反馈值对模型的影响；文章还进一步地在模型中引入基于假设检验的反馈检测方法，从而能够过滤恶意反馈。文章通过理

论分析，证明了模型的健壮性。实验结果表明 RLM 信誉模型能够有效地跟踪评估信誉的估计值和信誉估计方差。与简单相加模型和 Bayesian 模型相比，RLM 模型能够给出更准确的信誉值评估。在恶意反馈攻击的情况下，RLM 模型在被测试的模型中能够给出最小的信誉值估计误差；且在恶意反馈检测方面，RLM 模型比 Bayesian + Quantile 模型有着更小的假阳性率和更高的真阳性率。

基于策略和基于信誉的两种信任关系具有很强的互补性，然而目前的大多数的信任管理系统将两者分离，单独进行信任关系处理，它们无法提供全面的信任评估功能，如细粒度和动态的信任管理。本文介绍了一个基于角色策略和信誉的混合信任管理系统 RTE。RTE 的信任策略语言通过在基于角色的信任关系语言中增加信誉值参数，从而能够在信任策略语言中支持信誉的管理。RTE 的信誉值的计算包括信任经验和信任推荐，能够实现资源的细粒度访问控制授权。另外，RTE 的策略语言通过定义信任合成算子，能够支持信誉值的网络传递和计算，进而可以根据角色的跟踪记录，动态管理角色授权，抗击角色域内的恶意行为。文章给出了 RTE 策略语言的语法和推演规则，介绍了 RTE 系统的信任值计算，并给出了一个 RTE 系统进行混合信任管理的示例。

当前基于信任的应用都比较简单，缺少大规模的应用示范。另外目前的信任模型难以支持复杂应用系统的可靠性信任量化评估。为此，本文对复杂的网络作业流中信任的应用进行研究，提出一种基于信誉的面向可靠性的作业流的调度系统。为了评估资源的可靠性，文章提出了时间相关的可靠性驱动 RD 信誉模型。RD 信誉使用资源的作业失败率来定义信誉，从而使调度系统可以直接使用 RD 信誉来评估一个作业的可靠性。另外文章设计的 RD 信誉算法能够实时监控资源信誉状态的变化。基于 RD 信誉，本文提出了一个前瞻的遗传调度算法 LAGA 来智能地为工作流任务同时优化时间和可靠性信任。LAGA 采用我们提出的资源优先级启发式对传统遗传算法的变异算子进行优化，并使用一种新颖的演化和评估机制：遗传算子只负责演化调度方案的资源作业映射，而调度方案的作业顺序由算法的评估步骤采用我们提出的 max-min 策略进行智能决策。实验结果显示 RD 信誉能够计算出更准确的资源信誉，并能在工作流调度中增加一个工作流任务调度的可靠性。LAGA 算法能够给出比基于列表启发式 DLS 和 RDLS 更好的调度方案，而且，它比另一种遗传算法 BGA 有着更好的调度方案演化性能。

7.2 未来工作

围绕着我们提出的四个研究问题，本文提出了系统的解决方案。未来我们可以在已有工作的基础上，进一步深化信任管理和应用的研究，有待研究的问题简

述如下。

针对基于策略和信任证书的信任管理，当前很多应用需要在限制信任证书输出的情况下进行信任协商，本文的信任证书释放规则提供了信任证书输出限制的功能描述。下一步工作中，我们将在 RTP 语言和 DPN 算法的基础上，增加对信任证书加密技术支持，如隐藏信任证书和基于属性的加密技术等。另外信任的分布式证明方法在信任安全上是对传统信任协商方法的放松，保证信任证明的一致性成为信任协商系统很重要的因素，我们将开展这方面的研究。

针对基于信誉的信任模型，本文提出了一个健壮的通用信誉模型 RLM。下一步工作中，我们将主要考虑如何减小信誉反馈噪音对 RLM 信誉模型的影响。目前，反馈的噪音将会导致模型性能的退化。此外，我们将基于现有的统计推测理论，在 RLM 信誉模型的基础上增加健壮性的信誉评估技术，从而能够防范恶意信誉反馈攻击。

针对基于策略和信誉的混合信任管理，本文提出了一个支持信誉计算的角色管理系统 RTE。下一步的工作中，我们将基于 RTE 的策略语言和 ABE 技术，将信任值作为属性，为用户分配密钥，实现数据的分布式安全存储。另外目前很少有工作对混合信任的逻辑基础[80,81]及本体语义[48]进行阐述，这可能造成信任概念的模糊使用。下一步我们将开展这方面的研究。

针对信任概念的应用，本文提出了一个基于信誉的面向可靠性的工作流调度系统。但本文的工作流是一个以计算为主的应用，没有考虑作业之间的通信时间。为了应对当前以数据为中心的应用变化，下一步我们将研究如何使 RD 信誉以及我们的调度算法更好的反应分布式数据传输特性。

致　谢

爱、乐、静！

博士论文付梓之际，我五年多的研究生求学生涯即将结束。激动的心情把我的思绪带回到小学的操场、中学的教室和那些在大学实验室中度过的日日夜夜。回首求学路上的点点滴滴，我的每一步成长都离不开老师、亲人和朋友的关心与帮助，衷心感谢你们！

首先感谢我的导师苏金树教授！从本科开始，我就能够有幸师从苏老师的指导，这让我获益终身。忘不了本科阶段，老师耐心的听我那个晦涩难懂的技术报告，从您鼓励的眼神中，我得到了研究的乐趣和力量；忘不了硕士阶段，老师在新加坡参加国际会议的过程中，给我发的资料短信，正是由于老师不断的关心和指导，我才能在一条正确的研究道路上不断进步；忘不了博士阶段，老师提醒我要考好 PETS5 要勇敢出去闯，您对我的不断鞭策和鼓励，让我得以有更广阔的视野和更高的追求。老师多年的教诲，让我从一个涩涩青年成为如今能够有一技之长的科研后备军。老师高深的学术造诣、渊博的知识、严谨的治学作风以及努力的工作精神都将是我今后学习的源泉，并鞭策我一直向前，成长进步。再次谢谢您，苏老师！

感谢墨尔本大学的 Rajkumar Buyya 教授，一次交流会上的相识成就了我们一段师生情谊。为了让我能够顺利抵达墨大，您费尽周折，无私帮助。在墨尔本期间，您把我当成自己的博士生指导，亲历亲为。您为我改文章，总是强调参考文献格式要严整，致谢要严格。您在让我尊重别人劳动的同时，也让我更加体会到您待人的友善。作为国际知名学者，从你这儿我也深刻认识到开放、交流的要义和重要性。

感谢网络所的卢泽新、孙志刚、吴纯清、王勇军、胡晓峰、陈署晖、时向泉、钟求喜和赵峰等老师。各位老师作为科研一线的领导或骨干，为我们学生搭建了很好的研究平台和环境。你们深厚的学术积累和言传身教，也让我能够更快的成长进步。特别感谢孙志刚老师在我出国期间给我的关心，让我感受到组织的温暖。另外还要感谢佐治亚大学的 Ling Liu 教授，您对我论文的修改和建议让我知道精益求精的重要性。

感谢师兄（姐）张博峰，王圣，孟兆伟，赵宝康，曾迎之，黄清元，涂瑞，戴艺，曹继军，陈峰。你们的报告，让我学习了很多；你们对我课题的指导和意见，给了我很多启发和帮助。还要感谢张一鸣师兄，你帮我改文章的认真态度让我自觉惭愧，对我的帮助总是尽心尽力，非常感谢。

感谢墨尔本大学的 Chee Shin Yeo，Saurabh Garg，Rajiv Ranjan，Suraj Pandey，Marco A. S. Netto 等好友。正是由于你们的帮助，我在墨尔本的生活和工作才能如此惬意和丰实。特别是 Yeo，你像大哥一样关心我的生活和工作，我都当你是中国人了。另外，跟 Saurabh 讨论哲学问题很有意思。

感谢师门的马延鹏，胡乔林，孙一品，戴斌，白冰，冯振乾，曹丹，陆华彪。和你们一起学习、讨论问题真的很有意思。还要感谢师门的硕士师弟们，你们让我有做得更好的压力，后浪推前浪，胜在沙滩上。

感谢伴我寒窗苦读的同学们，在这就不逐一写名字了。从本科的 2000 级到硕士的 04 级以及博士的 06 级，我们共同走过了那些军训和考试的日子，也一起把酒当歌。这份同学情战友情将终身珍藏。还要感谢校微软技术俱乐部 04-07 的兄弟姐妹们，年轻的梦想和激情就是你我共有的财富。

感谢国防科技大学研究生院和计算机学院的参谋老师们，为了我们的学习、出国和毕业的事情，真是辛苦你们了。感谢学员大队各级领导多年来对我的培育和关心。感谢国家留学基金委对我联合培养的支持。也要感谢母校江苏省海安高级中学以及海安李堡中心初中对我的培养，根基对大树的作用不言而喻。

感谢女友邓月霞多年来对我的照顾和忍让。和你一起度过的美好时光让我懂得爱得付出和回报。感谢你宽容我的责备和无理情绪，你用你那特有的平静抚慰了我的焦躁。

最后把我所有的感激都送给我的家人。正如致谢的位置，你们永远是我感恩心情的汇聚点。你们无私伟大的爱，伴随我走遍天涯海角，使我永远不会感到孤独。感谢母亲用她的开朗和幽默，让我学会了乐此不疲的意境。感谢弟弟长期以来对我的支持，兄弟俩小时的那些“打斗”和“并肩冲杀”成了我有生以来回忆最多的画面。自从我出外求学后，跟家人的感情联络少了，这让我很是过意不去。尤其是对弟弟的帮助，难说尽了为哥之责。家人对我的爱无以为报，唯有在以后的日子里，多一份对家的体贴，多一份向你们汇报的成绩。

参考文献

[1] R.C. Mayer, J.H. Davis, D.F. Schoorman, An Integrative Model of Organizational Trust [J]. The Academy of Management Review, 1995, 20(3): 709-734.

[2] J. Huang and M. Fox. An ontology of trust: formal semantics and transitivity [C]. in Proceedings of the 8th international conference on Electronic commerce. 2006: ACM New York, NY, USA.

[3] D. Gambetta, Can We Trust Trust [J]? Trust: Making and Breaking Cooperative Relations, 1990, 213-238.

[4] T. Grandison and M. Sloman, A Survey of Trust in Internet Applications [J]. IEEE Communications Survey and Tutorials, 2000.

[5] F.L. Mayer, A brief comparison of two different environmental guidelines for determining 'levels of trust' [C], Sixth Annual Computer Security Applications Conference. 1990.

[6] E. Chang, T. Dillon, F. Hussain, The fuzzy and dynamic nature of trust [J]. LNCS 3592, 2005, 161-174.

[7] L. Zucker, Production of trust: Institutional sources of economic structure [J]. Research in Organizational Behavior, 1986, 8:53-111.

[8] A. Jøsang, R. Ismail, and C Boyd. A Survey of Trust and Reputation Systems for Online Service Provision [J]. Decision Support Systems, 2007, 43(2):618-644.

[9] F. Azzedin and M. Maheswaran. Evolving and Managing Trust in Grid Computing Systems [C]. Canadian Conference on Electrical and Computer Engineering. 2002.

[10] A. Donovan and G. Yolanda, A survey of trust in computer science and the Semantic Web [J], Journal of Web Semantics, 2007, 5: 58-71.

[11] J.P Anderson, Computer Security Technology Planning Study [R]. ESD-TR-73-51, Hanscom AFB, Bedford, MA, 1972.

[12] ISO/IEC, Information technology-security Techniques-Evaluation Criteria for IT Security [S], Part 1: Introduction and General Model. 2nd ed. 2005.

[13] Trust Computing Group [J], TCG Architecture Overview. 2004.

[14] B. Gates. Trustworthy Computing [R], 2002.

[15] A. Avizienis, J. Laprie, Basic concepts and taxonomy of dependable and secure computting [J]. IEEE Trans Dependable Secure, 2004, 1(1): 11-33.

[16] 林闯,可信网络研究[J].计算机学报, 2005, 28(5):751-758.

[17] 王怀民,唐扬斌,尹刚,李磊.互联网软件的可信机理[J],中国科学 E 辑:信息科学. 2006, 36(10): 1156~1169.

[18] M. Blaze, J. Feigenbaum, and J. Lacy. Decentralized trust management [C]. In IEEE Symposium on Security and Privacy, 1996, pages 164–173.

[19] Jøsang, A. and N. Tran, Trust Management for E-Commerce [C], 2000. http://citeseer.nj.nec.com/375908.html

[20] T. Grandison, Trust Management for Internet Applications [D], PhD Thesis, 2003

[21] A. Abdul-Rahman, S. Hailes, A distributed trust model [C], In: Proceedings of the 1997 New Security Paradigms Workshop. Cumbria, UK, ACM Press, 1998: 48~60.

[22] D. Povey, Developing electronic trust policies using a risk management model [C].In: Proceedings of the 1999 CQRE Congress. 1999.

[23] Yu T, M. Winslett, A unified scheme for resource protection in automated trust negotiation [C]. In: Proc. of the 2003 IEEE Symp. On Security and Privacy. Washington: IEEE Computer Society Press, 2003.

[24] Johnson W, Mudumbai S, Thompson M. Authorization and attribute certificates for widely distributed access control [C]. In: IEEE Proc. of the 7th Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises. Washington: IEEE Computer Society Press, 1998. 340−345.

[25] Leithead T, Nejdl W, Olmedilla D, Seamons KE, Winslett M, Yu T, Zhang C. How to exploit ontologies in trust negotiation [C]. In: Workshop on Trust, Security, and Reputation on the Semantic Web, Part of the 3rd Int'l Semantic Web Conf. 2004.

[26] Yu T, Ma X, Winslett M. PRUNES: An efficient and complete strategy for trust negotiation over the Internet [C]. In: Proc. of the 7th ACM Conf. on Computer and communications Security. New York: ACM Press, 2000. 210−219.

[27] Barlow T, Hess A, Seamons KE. Trust negotiation in electronic markets [C]. In: Proc. of 8th Research Symp. in Emerging Electronic Markets. Maastricht, 2001.

[28] Bosworth KP, Tedeschi N. Public key infrastructures: The next generation [J]. Journal of BT Technology, 2001,19(3):44−59.

[29] HAYTON, R. J., J. M. BACON, ET AL. Access Control in an Open Distributed Environment [C]. IEEE Symposium on Security and Privacy, 1998.

[30] Thompson MR, Essiari A, Mudumbai S. Certificate-Based authorization policy in a PKI environment [C]. In: Thompson MR, ed. Proc. of the Information and System Security. New York: ACM Press, 2003.

[31] Li LX, Chen WM, Huang SL. Realizing mandatory access control in role-based security system [J]. Journal of Software, 2000,11(10):1320-1325 (in Chinese with English abstract).

[32] Saunders G, Hitchens M, Varadharajan V. Role-Based access control and the access control matrix [C]. In: Hitchens M, ed. Proc. of the ACM SINGOPS Operating Systems Review. New York: ACM Press, 2001.

[33] Zhang LH, Ahn GJ, Chu BT. A rule-based framework for role-based delegation [C]. In: Proc. of the 6th ACM Symp. on Access Control Models and Technologies. New York: ACM Press, 2001.

[34] Zhang XZ, Oh S, Sandhu R. PBDM: A flexible delegation model in RBAC [C]. In: Ferrari E, Ferraiolo D, eds. Proc. of the 8th ACM Symp. on Access Control Models and Technologies. New York: ACM Press, 2003.

[35] Seamons KE, Winslett M, Yu T. Limiting the disclosure of access control policies during automated trust negotiation [C]. In: Network and Distributed System Security Symp (NDSS 2001). Internet Society Press, 2001. http://isrl.cs.byu.edu/pubs/ndss2001.pdf

[36] Seamons KE, Winslett M, Yu T, Smith B, Child E, Jacobson J, Mills H, Yu L. Requirements for policy languages for trust negotiation [C]. In: Michael JB, ed. Proc. of the 3rd IEEE Int'l Workshop on Policies for Distributed Systems and Networks. Washington: IEEE Computer Society Press, 2002.

[37] Winsborough WH, Li NH. Towards practical automated trust negotiation [C]. In: the 3rd Int'l Workshop on Policies for Distributed Systems and Networks. Washington: IEEE Computer Society Press, 2002. 92−103.

[38] Yu T, Winslett M, Seamons KE. Interoperable strategies in automated trust negotiation [C]. In: Proc. of the 8th ACM Conf. on Computer and Communications Security. New York: ACM Press, 2001. 146−155.

[39] Yu T, Winslett M, Seamons KE. Supporting structured credentials and sensitive policies through interoperable strategies for automated trust negotiation [J]. ACM Trans. on Information and System Security, 2003,1(6):1−42.

[40] M. Blaze, J. Feigenbaum, J. Ioannidis, and A. D. Keromytis. RFC 2704: The KeyNote trust management system version2 [S], 1999.

[41] C. M. Ellison, B. Frantz, B. Lampson, R. Rivest, B. M. Thomas, T. Ylonen. SPKI Certificate Theory [S]. RFC 2693, 1999.

[42] Chu, Y.-H., Feigenbaum, J., LaMacchia, B., et al． REFEREE: trust management for Web applications [J]. World Wide Web Journal, 1997, 2(2): 127-139

[43] P. Resnick, R. Zeckhauser, E. Friedman, and K. Kuwabara, ReputationSystems [J], Communications of the ACM, 43(12), December 2000, pp.45-48

[44] Abdul-Rahman A., Hailes S., Supporting trust in virtual communities [C], in Proceedings of International Conference on System Sciences, Hawaii. 2000.

[45] Mui L., Mohtashemi M., Halberstadt A., A Computational Model of Trust and Reputation for E-businesses [C], in Proceedings of the 35th Annual International Conference on System Sciences (HICSS'02)-Volume 7, Hawaii, IEEE ComputerSociety, Washington, DC, USA. 2002.

[46] Chang E., Dillon T., Hussain F.K., Trust and Reputation for Service-Oriented Enviornments: Technologies for Building Business Intelligence and Consumer Confidence, John Wiley & Sons. 2005.

[47] Sandhu, R. S., E. J. COYNE, ET AL. Role- Based Access Control Models [J].

IEEE Computer 29(2), 1996. 38−47.
[48] A. Mohan, D.M. Blough, AttributeTrust A Framework for Evaluating Trust in Aggregated Attributes via a Reputation System [C], Sixth Annual Conference on Privacy, Security and Trust (PST), 2008.
[49] S. Chakraborty and I. Ray, TrustBAC Integrating Trust Relationships into the RBAC Model for Access Control in Open Systems [C], SACMAT 2006
[50] J. Huang and D. Nicol, A Calculus of Trust and Its Application to PKI and Identity Management [C], Proceedings of the 8th Symposium on Identity and Trust on the Internet, 2009.
[51] Winsborough WH, Seamons KE, Jones VE. Automated trust negotiation [C]. In Hilton SC, eds. Proc. of the DARPA Information Survivability Conf. and Exposition. New York: IEEE Press, 2000.
[52] Winslett M, Zhang C, Bonatti PA. PeerAccess: A logic for distributed authorization [C]. In Atluri V, Meadows C, Juels A Eds. Proc. of the ACM Conf. on Computer and Communications Security. New York: ACM Press, 2005.
[53] Lee AJ, Winslett M. Safety and consistency in policy-based authorization systems [C]. In Lin FC, Lee DT, Lin BS, Shieh S, Jajodia S Eds. Proc. of the ACM Conf. on Computer and Communications Security. New York: ACM Press, 2006.
[54] Bauer L, Garriss S, Reiter MK. Distributed proving in access-control systems [C]. In Paxson V, Waidner M eds. Proc. of the IEEE Symp. on Security and Privacy. Washington: IEEE Computer Society Press, 2005.
[55] Li NH, Winsborough WH, Mitchell JC. Distributed credential chain discovery in trust management [C]. In: Herbert AS, eds. Proc. of the 8th ACM Conf. on Computer and Communications Security. New York: ACM Press, 2001.
[56] Li J, Li N, Winsborough WH. Automated trust negotiation using cryptographic credentials [C]. In Atluri V, Meadows C, Juels A Eds. Proc. of the ACM Conf. on Computer and Communications Security. New York: ACM Press, 2005.
[57] Irwin K, Yu T, Winsborough WH. On the modeling and analysis of obligations [C]. In Lin FC, Lee DT, Lin BS, Shieh S, Jajodia S Eds. Proc. of the ACM Conf. on Computer and Communications Security. New York: ACM Press, 2006.
[58] Stoller SD, Yang P, Ramakrishnan CR, Gofman MI. Efficient policy analysis for administrative role based access control [C]. In Ning P, Vimercati S, Syverson PF Eds. Proc. of the ACM Conf. on Computer and Communications Security. New York: ACM Press, 2007.
[59] Skogsrud H, Benatallah B, Casati F. Trust-Serv: Model-Driven lifecycle management of trust negotiation policies for web services [C]. In Feldman SI, Uretsky M, Najork M, Wills CE Eds. Proc. of the 13th Int'l World Wide Web Conf. (WWW 2004). New York: ACM Press, 2004.
[60] Bertino E, Ferrari E, Squicciarini AC. Trust-X: A peer-to-peer framework for trust

establishment [J]. IEEE Trans. on Knowledge and Data Engineering, 16(7):827-842, 2004.

[61] Liao ZS, Jin H, Li CS, Zou DQ. Automated trust negotiation and its development trend [J]. Journal of Software, 17(9): 1933-1948, 2006.

[62] Li JX, Huai JP, Li XX. Research on automated trust negotiation [J]. Journal of Software, 2006, 17(1):124-133.

[63] Bradshaw R, Holt J, Seamons K. Concealing complex policies with hidden credentials [C]. Proc. of the 11th ACM Conf. on Computer and Communications Security. New York: ACM Press, 2004.

[64] Li J, Li N. OACerts: Oblivious attribute certificates [J]. IEEE Trans. on Dependable and Secure Computing, 2006, 3(4):340-352.

[65] Li NH, Mitchell JC, Winsborough WH. Design of a role-based trust management framework [C]. In Heather H, eds. Proc. of the IEEE Symp. on Security and Privacy. Washington: IEEE Computer Society Press, 2002.

[66] Li NH, Mitchell JC. Datalog with constraints: A foundation for trust management languages [C]. In Verónica D, Philip W, eds. Proc. of the Int'l Symp. on Practical Aspects of Declarative Languages. LNCS 2562, Berlin, Heidelberg: Springer-Verlag, 2003.

[67] Minami K, Kotz D. Scalability in a secure distributed proof system [C]. In: Proc. of the Int'l Conf. on Pervasive Computing. Berlin: Springer, 2006.

[68] Li JX, Huai JP. COTN: A contract_based trust negotiation system [J]. Chinese Journal of Computers, 29(8): 1290-1300, 2006. (in Chinese with English abstract).

[69] Sergio Marti and Hector Garcia-Molina, Taxonomy of trust: Categorizing P2P Reputation Systems [J], Computer Networks 2006, 50(4): 472-484.

[70] J. Douceur. The sybil attack [C]. In Proceedings of the IPTPS02 Workshop, Cambridge, MA (USA), 2002.

[71] M. Gupta, P. Judge, M.H. Ammar, A reputation system for peer-to-peer networks [C], in: ACM NOSSDAV, 2003.

[72] K. Aberer, Z. Despotovic, Managing Trust in a Peer-2-Peer Information System [C], in Proceedings of Intl. Conf. on Information and Knowledge Management. 2001.

[73] Amazon. (Available at: http://www.amazon.com/, last accessed on Aug.10, 2009).

[74] R. Chen, W. Yeager, Poblano:A Distributed Trust Model for P2P Networks [R], Sun Microsystem, Palo Alto, Tech Rept:TR-I4-02-08, 2002.

[75] Lintao Liu, Shu Zhang, Kyung Dong Ryu, and Partha Dasgupta: R-Chain: A Self-Maintained Reputation Management System in P2P Networks [C]. 17th International Conference on Parallel and Distributed Computing Systems (PDCS-2004), September 2004, San Francisco, CA, USA, pages 131-136.

[76] Sergio Marti,Hector Garcia-Molina.Limited reputation sharing in p2p systems [C], In Proceedings of the 5th ACM Conference on Electronic Commerce.New

York,NY,USA:ACM Press,2004.91-101.

[77] eBay. (Available at: http://www.ebay.com, last accessed on Dec.10, 2006).

[78] K.Aberer, P-Grid: A Self-Organizing Access Structure for P2P Information Systems [C], Sixth International Conference on Cooperative Information Systems (CoopIS 2001), Trento, Italy, LNCS 2172, Springer-Veralg, 2001,pp.179-194.

[79] S. Ratnasamy et al., A Scalable Content-Addressable Network [C], Proceeding of ACM SIGCOMM,ACM Press, NewYork, 2001.8, pp.161-172.

[80] A. Shirazi and E. Amir, Probabilistic Modal Logic [C], Int. Conference on Artificial Intelligence (AAAI), 2007.

[81] Z. Cao and C. Shi, Probabilistic Belief Logic and Its Probabilistic Aumann Semantics [J], Journal of Computer Science and Technology,2003, 18(5):571-579.

[82] J. Golbeck. Weaving a Web of Trust [J]. Science, 2008, 321(5896): 1640 – 1641.

[83] P. Resnick, R. Zeckhauser, J. Swanson, and K. Lockwood. The Value of Reputation on eBay: A Controlled Experiment [J]. Experimental Economics, 2006, 9(2): 79-101.

[84] D. Stern, R. Herbrich and T. Graepel, Matchbox: Large Scale Online Bayesian Recommendations [C], 18th international conference on World Wide Web (WWW), 2009.

[85] L. Xiong and L. Liu, PeerTrust: Supporting Reputation-Based Trust for Peer-to-Peer Electronic Communities [J], IEEE Trans. Knowledge and Data Eng, 2004, 16(7): 843-857.

[86] R. Zhou and K. Hwang, PowerTrust: A Robust and Scalable Reputation System for Trusted Peer-to-Peer Computing [J], IEEE Trans. on Parallel and Distributed Systems, 2006, 18(5): 460-473.

[87] S. Song, K. Hwang, and Y.K. Kwok, Risk-Resilient Heuristics and Genetic Algorithms for Security-Assured Grid Job Scheduling [J], IEEE Trans. on Computers, 2006, 55(6):703-719.

[88] J. Weng, C. Miao and A. Goh, A Robust Reputation System for the Grid [C], IEEE/ACM International Symposium on Cluster Computing and the Grid, 2006.

[89] A. Whitby, A. Jøsang, and J. Indulska. Filtering out unfair ratings in bayesian reputation systems [C]. In Proceedings of the Workshop on Trust in Agent Societies at AAMAS, 2004.

[90] Y. Wang and M. P. Singh, Trust representation and aggregation in distributed agent systems [C], Int. Conference on Artificial Intelligence (AAAI), Boston, 2006.

[91] B. Yu and M. P. Singh, Detecting Deception in Reputation Management [C], Int. Joint Conference on Autonomous Agents and Multiagent Systems(AAMAS), 2003.

[92] U. Kuter and J. Golbeck, SUNNY: A New Algorithm for Trust Inference in Social Networks Using Probabilistic Confidence Models [C], Int. Conference on Artificial Intelligence (AAAI), 2007.

[93] J. Golbeck and J. Hendler, Inferring Trust Relationships in Web-based Social Networks [J], ACM Transaction on Internet Technology, 6(4): 497-529, 2006.
[94] M. Raya, P. Papadimitratos, V.D. Gligor, J.P. Hubaux, On DataCentric Trust Establishment in Ephemeral Ad Hoc Networks [C], In Proceedings of IEEE Infocom, 2008.
[95] K. Krukowy, M. Nielsen and V. Sassonex, A Framework for Concrete Reputation-Systems with Applications to History-Based Access Control [C], In Proceedings of ACM CCS, 2005.
[96] M. Srivatsa, L. Xiong and L. Liu, TrustGuard: countering vulnerabilities in reputation management for decentralized overlay networks [C], 14th international conference on World Wide Web (WWW), May 2005.
[97] J.M. Morris, The Kalman filter: A robust estimator for some classes of linear quadratic problems [J], IEEE Transactions on Information Theory, 22(5): 526–534, 1976.
[98] C. Atkeson, A. Moore and S. Schaal. Locally weighted learning [J]. AI Review, 11:11–73, April 1997.
[99] P.S. Maybeck, Stochastic models, estimation, and control [J]. Mathematics in Science and Engineering, Volume 141, Academic Press, 1979
[100] A. Dempster, N. Laird and D.Rubin, Maximum likelihood from incomplete data via the EM algorithm [J]. Journal of Royal Statistical Society. Series B 39(1): 1–38, 1977.
[101] M.A. Kaafar, L. Mathy, C. Barakat et. al , Securing Internet Coordinate Embedding Systems [C], in Proceedings of the ACM SIGCOMM, August 2007.
[102] J. Ting, A. D’Souza and S. Schaal, Bayesian regression with input noise for high dimensional data [C], ACM Proceedings of the 23rd International Conference on Machine Learning, 2006.
[103] F. Angiulli, S. Basta, and C. Pizzuti, Distance-Based Detection and Prediction of Outliers [J], IEEE Transactions on Knowledge and Data Engineering, 18(2): 145-160, 2006.
[104] Y.L. Sun, Z. Han, W. Yu and K.J.R. Liu, A Trust Evaluation Framework in Distributed Networks: Vulnerability Analysis and Defense Against Attacks [C], In Proceedings of IEEE Infocom'06, 2006.
[105] K. Walsh and E. G. Sirer. Experience with an object reputation system for peer-to-peer filesharing [C]. In USENIX Symposium on Networked Systems, Design and Implementation (NSDI), 2006.
[106] W.Y. Chen, D. Zhang and E.Y. Chang, Combinational Collaborative Filtering for Personalized Community Recommendation [C], ACM International Conference on Knowledge Discovery and Data Mining, 2009.
[107] E. Damiani, D.C. di Vimercati, S. Paraboschi, P. Samarati, and F. Violante, A

reputation-based approach for choosing reliable resources in peer-to-peer networks [C], in Proceedings ACM CCS, 2002.

[108] S. Kamvar, M. Schlosser, and H. Garcia-Molina, The Eigentrust Algorithm for Reputation Management in P2P Networks [C], 12th international conference on World Wide Web (WWW), May 2003.

[109] Y. Zhang and Y. Fang, A Fine-Grained Reputation System for Reliable Service Selection in Peer-to-Peer Networks [J], IEEE Trans. on Parallel and Distributed Systems, 18(8): 1134 - 1145, 2007.

[110] S. Rubin, M. Christodorescu, V. Ganapathy, J.T. Giffin, L. Kruger, H. Wang, N. Kidd, An auctioning reputation system based on anomaly [C], Proceedings of ACM CCS, 2005.

[111] C. Dellarocas. Immunizing Online Reputation Reporting Systems Against Unfair Ratings and Discriminatory Behavior [C]. In ACM Conference on Electronic Commerce, Minneapolis, MN, USA, 2000: 150-157.

[112] M. Chen and J. Singh. Computing and Using Reputations for Internet Ratings [C]. In Proceedings of the Third ACM Conference on Electronic Commerce (EC'01). ACM, October 2001, 154-162.

[113] Wang Y., Vassileva J., Bayesian Network-Based Trust Model in P2P Networks [C], in Proceedings of AP2PC 2003, IEEE Computer Society. 2003. p.372-378.

[114] Z. Despotovic, K. Aberer, Maximum Likelihood Estimation of Peers' Performances in P2P Networks [C], in: 2nd Workshop on the Economics of Peer-to-Peer Systems, Cambridge, MA, USA, 2004.

[115] D.W. Manchala: Trust metrics, models and protocols for electronic commerce transactions [C]. In The 18th International Conference on Distributed Computing Systems, 3-12, 1998.

[116] L. Page, S. Brin, R. Motwani, and T. Winograd. The PageRank Citation Ranking: Bringing Order to the Web. Technical report [J], Stanford Digital Library Technologies Project, 1998.

[117] R. Levien. Attack Resistant Trust Metrics [D]. PhD thesis, University of California at Berkeley, 2004.

[118] A. Jøsang. A Logic for Uncertain Probabilities. International Journal of Uncertainty [J], Fuzziness and Knowledge-Based Systems, June 2001, 9(3):279-311.

[119] B.T. Adler and L. Alfaro, A Content-Driven Reputation System for the Wikipedia [C], WWW 2007.

[120] K. Hoffman, D. Zage and C. Nita-Rotaru, A Survey of Attack and Defense Techniques for Reputation Systems [J], ACM Computing Surveys. 14(4), 2009.

[121] J. Sonnek, A. Chandra, and J. Weissman. Adaptive Reputation-Based Scheduling on Unreliable Distributed Infrastructures [J]. IEEE Transactions on

Parallel and Distributed Systems, 18(11):1151-1564, 2007.

[122] 张强,基于信任机制的分簇 MANET 关键技术研究[D],博士毕业论文,国防科学技术大学,2009.

[123] I. Foster, and A. Iamnitchi. On Death, Taxes, and the Convergence of Peer-to-Peer and Grid Computing [C]. 2nd Int'l. Workshop on P2P Systems, 2003.

[124] D. Lima, Y. Onga, Y. Jinb, B. Sendhoffb, and B. Lee, Efficient Hierarchical Parallel Genetic Algorithms using Grid computing [J], Future Generation Computer Systems, 2007, 23(4):658-670.

[125] A. Sulistio, G. Poduval, R. Buyya, and C. Tham, On Incorporating Differentiated Levels of Network Service into GridSim [J], Future Generation Computer Systems (FGCS) , 2007, 23(4):606-615.

[126] S.C. Kim, S. Lee, and J. Hahm, Push-Pull: Deterministic Search-Based DAG Scheduling for Heterogeneous Cluster Systems [J], IEEE Trans. on Parallel and Distributed Systems, 2007, 18(11):1489-1052.

[127] T. D. Braun, H. J. Siegel, N. Beck et al, A comparison of eleven static heuristics for mapping a class of independent tasks onto heterogeneous distributed computing systems [J], J. of Parallel and Distributed Computing, 2001, 61(6):810-837.

[128] J. Yu, M. Kirley, and R. Buyya, Multi-objective Planning for Workflow Execution on Grids [C], IEEE/ACM Conference on Grid Computing, 2007.

[129] 尹刚,域间计算环境中授权管理研究与实现[D],博士毕业论文,国防科学技术大学,2006.

[130] 曲向丽,网格环境下互信机制关键技术研究[D],博士毕业论文,国防科学技术大学,2006.

[131] 常俊胜,虚拟计算环境下基于信誉的信任管理研究[D],博士毕业论文,国防科学技术大学,2006.

[132] 唐扬斌,虚拟计算环境下激励相容的信誉机制研究[D],博士毕业论文,国防科学技术大学,2006.

[133] M. Wieczorek, S. Podlipnig, R. Prodan, and T. Fahringer. Bi-criteria Scheduling of Scientific Workflows for the Grid [C]. IEEE International Symposium on Cluster Computing and the Grid, 2008.

[134] A. Benoit, M. Hakem, and Y. Robert. Fault tolerant scheduling of precedence task graphs on heterogeneous platforms [C]. IEEE International Symposium on Parallel and Distributed Processing (IPDPS), 2008.

[135] J Dongarra, E Jeannot, E Saule and Z Shi. Bi-objective Scheduling Algorithms for Optimizing Makespan and Reliability on Heterogeneous Systems [C]. ACM Symposium on Parallelism in Algorithms and Architectures, 2007.

[136] M. Hakem, F. Butelle, Reliability and Scheduling on Systems Subject to Failures [C]. International Conference on Parallel Processing (ICPP), Sept. 2007.

[137] A. Dogan and F. Ozguner. Bi-objective Scheduling Algorithms for Execution Time-Reliability Trade-off in Heterogeneous Computing Systems [J]. The Computer Journal, 2005. 48(3):300-314.

[138] A. Dogan and F. Ozguner. Matching and scheduling algorithms for minimizing execution time and failure probability of applications in heterogeneous computing [J]. IEEE Trans. on Parallel and Distributed Systems, 2002, 13(03):308-323.

[139] I. Assayad, A. Girault, and H. Kalla, A bi-criteria scheduling heuristic for distributed embedded systems under reliability and real-time constraints [C], in Proc. of the IEEE DSN, Florence, Italy, Jun, 2004.

[140] X. Qin, T. Xie, An Availability-Aware Task Scheduling Strategy for Heterogeneous Systems [J], IEEE Transactions on Computers, 2008, 57(2):188-199.

[141] S. Zhao and V. Lo, Result Verification and Trust-based Scheduling in Open Peer-to-Peer Cycle Sharing systems [C], in IEEE Fifth International Conference on Peer-to-Peer Systems, Sept. 2005.

[142] D. Kondo, G. Fedak, F. Cappello, A.A. Chien, and H. Casanova: Characterizing resource availability in enterprise desktop grids [J]. Future Generation Comp. Syst, 2007. 23(7):888-903.

[143] L. Wang, H. J. Siegel, V. P. Roychowdhury, et al., Task matching and scheduling in heterogeneous computing environments using a genetic-algorithm-based approach [J], J. of Parallel and Distributed Computing, 1997, 47(1):8-22.

[144] R.C. Corrêa, A. Ferreira, and P. Rebreyend, Scheduling Multiprocessor Tasks with Genetic Algorithms [J]. IEEE Trans. on Parallel and Distributed Systems, 1999. 10(8): 825-837.

[145] M. Hakem and F. Butelle. Critical path scheduling parallel programs on unbounded number of processors [J]. Int'l Journal of Foundations of Computer Science, 2006, 17(2):287–301.

攻读博士期间取得的学术成果

[1] Xiaofeng Wang, CheeShin Yeo, Rajkumar Buyya, Jinshu Su, Reliability-Driven Reputation Based Scheduling for Public-Resource Computing Using GA [C], Proceeding of the 23rd IEEE International Conference on Advanced Information Networking and Applications (AINA). Bradford, UK, 2009. (EI 收录)

[2] Xiaofeng Wang, Rajkumar Buyya and Jinshu Su, Reliability-Oriented Genetic Algorithm for Workflow Applications Using Max-Min Strategy [C], Proceeding of the 9th IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGRID). Shanghai, China, 2009. (EI 收录,分布式计算权威会议)

[3] Xiaofeng Wang, Wei Ou, Jinshu Su, A Distributed Proving and Negotiation Algorithm for Trust Management [C], International Conference on Information Science and Engineering (ICISE), Nanjing, China, 2009. (EI 收录)

[4] Xiaofeng Wang, Wei Ou, Jinshu Su, A Reputation Inference Model Based on Linear Hidden Markov Process [C], International Conference on Communication Systems, Networks and Applications, Sanya, China, 2009. (EI 收录)

[5] 王小峰,苏金树,张强,张一鸣,面向分布式证明的信任协商策略语言和方法[J], 软件学报, http://www.jos.org.cn/1000-9825/3491.htm, 2009.

[6] 王小峰,马延鹏,苏金树,一种开放网络环境下的分布式信任证明算法[J],计算机工程与科学,将发表于 2010 年 第 06 期.

[7] 王小峰,时向泉,苏金树,一种 TCP/IP 卸载的数据零拷贝传输方法[J],计算机工程与科学,2008 年 第 02 期.

[8] Xiaofeng Wang, Ling Liu, Jinshu Su, Yiming Zhang, RLM: A Comprehensive and Robust General Model for Trust Representation and Aggregation, Submitted.

[9] 苏金树,王小峰,欧嵬,曹丹,可靠性驱动的资源信誉管理和工作流前瞻遗传调度算法,已投.

[10] 王小峰,张博锋,胡乔林,苏金树,一种基于线性隐马尔科夫过程的通用信誉模型,已投.

[11] Wei Ou, Xiaofeng Wang, Wenbao Han, Yongjun Wang. Research on Trusted Network Model Based on BLP Model. ICCIT 2009. (EI 收录)

[12] Wei Ou, Xiaofeng Wang, Wenbao Han, Yongjun Wang. Research on Trust

Evaluation Model Based on TPM. IFSN 2009. (EI 收录)

[13] 张强,王小峰,龚正虎,基于角色信任的 MANET 认证机制与自动信任协商系统,通信学报.(已投稿)

[14] 冯冲,罗军,王小峰,陈迪,基于信任值的动态角色定义和管理语言,电子学报(增刊).

攻读博士期间参与的科研工作

[1] 某型高速报文传输接口设计项目,2006-2007,主要开发设计者.

[2] XXX 安全管理,国防预研项目,2006-2007,主要参与者.

[3] 新一代互联网体系结构理论研究,国家重点基础研究发展计划(973),2006-2008,主要参与者.

[4] QoS-based Scheduling of e-Research Application Workflows on Global Grids, Discovery Project, Australian Research Council, 2007- 2008,主要参与者.

[5] 新型互联网互联控制理论与方法研究(No.90604006),国家自然科学基金项目,2007-2009,主要参与者.