大连理工大学

硕士学位论文

基于角色和任务的网格安全需求分析模型

姓名：赵辉

申请学位级别：硕士

专业：软件工程

指导教师：李明楚

20061217

摘　要

网格是一个集成的计算与资源环境，即通过标准的、开放的、通用的协议和接口实现分布式资源的协同使用并且能够向用户提供非平凡的服务质量。网格环境下多用户参与的协同计算是网格的一个重要应用方向。网格应用的发展对于运行的可靠性提出了更高的要求。在网格环境中，安全协议是保证安全性的重要手段。网格环境下的安全需求包括鉴别，授权，证书的撤回，机密性，分布式信任，信息的完整性，不可抵赖性以及新鲜性。安全模型是基于系统安全需求建立的模型，用于描述系统的安全需求和安全体系的逻辑构成。现在安全设计都非常重视安全模型的构建。传统的安全模型很难适合于网格环境下多用户协同计算。

本文通过现有的安全模型的分析，提出了基于角色和任务的网格协同计算协同关系描述模型 TRBCR，而且在其基础上构建网格协同计算安全需求分析模型 TRBCC，实现了网格环境下多用户协同计算安全需求的形式化描述，从而把网格协同计算环境下的不同安全需求统一在一个理论体系中。

另外，本文通过引入协同计算信道的概念，在传统 Strand Space 理论的基础上提出了一种基于虚拟组织的网格安全协议形式化验证方法，实现了网格环境下多用户协同计算安全协议的形式化分析与证明。本文通过实例展示 TRBCC 网格安全需求分析模型以及扩展的 Strand Space 理论与传统安全模型的不同。在实例中，本文建立了跨虚拟组织的网格协同计算场景，利用 TRBCC 网格安全需求分析模型对其多用户协同计算安全需求进行了形式化描述，并且进一步通过基于虚拟组织的网格安全协议形式化验证方法对跨虚拟组织网格协同计算环境下的双向认证协议进行了验证。

关键词：网格；协同计算；安全模型；安全协议；形式化分析

A Grid Security Requirement Analysis Model Based on Roles and Tasks

Abstract

Grid is an integrated computing and resource infrastructure that implements distributed resource coordinative operations and guarantees a certain level of QoS to users based on standardized, open and common protocols and interfaces. Multi-user coordinative computing in grid environment is an important research field. With the development of grid applications, higher system reliability is required. security protocol is an important method for maintenance of system security in grid environment, which has security requirements including authentication, authorization, revoke of certificates, confidentiality, distributed trust, integrity and non-repudiation. Security model is based on system security requirements which provide specification for system security requirements and logic structure of security architecture. Modern security system design is very concerned with the building of security model and traditional security model is not adapt to the multi-user coordinative computing in grid environment.

This paper proposes a multi-user coordinative relationship specification model TRBCR based on roles and tasks for grid computing. Additionally, a grid security requirement analysis model TRBCC(based on that) is defined to provide formal security requirement specifications so that different security requirements are unified into one single theory architecture for multi-user coordinative computing in grid environment.

In this paper, the notion of grid computing channel is introduced and a formal method based on the traditional Strand Space theory is defined to provide analysis and verification for grid security protocols of multi-user coordinative computing in grid environment. The difference between TRBCC model and traditional security model is demonstrated by an example in which multi-VO coordinative computing process is build up and the formal specification of multi-user coordinative computing requirements is performed by the TRBCC model. Additionally, authentication protocol in the multi-VO coordinative computing environment is verified by the extended Strand Space theory in the example.

Key Words: Grid; Coordinative Computing; Security Model; Security Protocol; Formal Analysis

独创性说明

作者郑重声明：本硕士学位论文是我个人在导师指导下进行的研究工作及取得研究成果。尽我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写的研究成果，也不包含为获得大连理工大学或者其他单位的学位或证书所使用过的材料。与我一同工作的同志对本研究所做的贡献均已在论文中做了明确的说明并表示了谢意。

作者签名：赵辉 日期：2006.12.19

大连理工大学学位论文版权使用授权书

本学位论文作者及指导教师完全了解“大连理工大学硕士、博士学位论文版权使用规定”，同意大连理工大学保留并向国家有关部门或机构送交学位论文的复印件和电子版，允许论文被查阅和借阅。本人授权大连理工大学可以将本学位论文的全部或部分内容编入有关数据库进行检索，也可采用影印、缩印或扫描等复制手段保存和汇编学位论文。

作者签名：赵辉

导师签名：李明君

2006 年 12 月 19 日

引　言

网格是一个集成的计算与资源环境，其通过网络连接地理上分布的计算机、数据库等各类计算资源，形成了广域范围的“无缝和集成的协同计算环境”，实现了分布式计算、高吞吐量计算、协同工程和数据查询等诸多功能。目前，网格应用范围涉及科学，军事和社会经济等领域，例如用于大规模军事仿真的 SF Express，模拟黑洞的 Cactus，海量数据处理的 DataGrid 等。网格已经发展成为连接和整合各类不同计算资源的一种基础设施。

网格应用的发展对于运行的可靠性提出了更高的要求。在网格环境中，安全协议是保证安全性的一种重要手段。网格计算环境下的安全需求包括鉴别，授权，证书的撤回，机密性，分布式信任，信息的完整性，不可抵赖性以及新鲜性。在所有安全协议中认证和授权构成了整个网格安全的基础。

安全需求分析模型是基于系统安全需求建立的模型，着重于描述系统安全需求的逻辑构成。现在安全设计都非常重视安全模型的构建。在确定安全模型后，可以更精确的把握住系统的安全需求，从而可以更有效的布置安全策略，设计安全协议。

安全模型的设计主要依靠安全建模技术来实现，即通过对系统安全需求的分析，建立系统的静态和动态安全需求的形式化描述。对系统安全需求进行形式化描述有助于理解协议设计目的。目前人们在安全协议中采用形式化方法进行研究时，主要工作集中在了协议安全性的验证方面。而在协议安全需求的形式化描述方面虽然已经做了一些工作，但此领域仍然很少涉及。

目前,在该领域的研究大致有以下两个方向:

(1) 建立通用的认证协议安全需求

Diffie、van Oorschot 和 Wiener[1]对认证协议安全需求进行了非形式化描述。他们认为，认证协议的会话密钥应该保持秘密，协议应该匹配运行。其中，协议匹配运行是指如果协议参与者 A 与 B 运行协议，那么 A 接收来自 B 的消息数据和 B 发送给 A 的消息数据要求匹配。Bellare，Rogaway[2]和 Syverson[3]分别对协议的匹配运行安全需求进行了形式化描述，并在此基础上提出各自的安全需求分析模型。目前，得到广泛认可的是 Woo 和 Lam[4]提出的认证协议的安全需求定义。Woo 和 Lam 认为认证协议的基本功能是认证和密钥分配,那么相应地，就具备两个基本安全属性：对应性和秘密性。对应性是指对于一个协议主体的事件发生，必定有相对的另一主体的相关事件发生，两个事件之间有严格的时间先后关系。对应性和协议的认证功能相对应。对应性和匹配运行的描述非常相似，却比匹配运行的应用范围更广泛，因为对应性中的关联事件并不仅指同一个

消息数据的发送和接收。

(2) 针对不同类型协议,建立不同的安全需求

Syverson 和 Meadows[5]在为 NRL 协议分析器所设计的语言中首次采用了针对不同类型的协议，建立不同的安全需求的方式。其安全需求定义和 Woo-Lam 模型中的对应性概念在形式上有一些相似，都基于事件的发生次序。但是，和 Woo-Lam 模型中对应性安全需求可应用于所有协议不同，语言使用者可针对不同协议定义不同安全需求。它们已经给出了针对各种消息认证协议和可重复认证密钥分配协议的安全需求形式化描述方法。

安全协议的分析验证则是对安全协议进行评价的一个重要方法。只有在对其进行分析验证之后，确认没有缺陷的安全协议才是可用的协议。

对应于网格环境的复杂性，网格安全问题错综复杂，为了满足各种安全需求，必须根据较为完善的安全模型来设计针对不同安全目标的安全协议。

对于传统网络环境下的协同计算，由于参与者和使用的资源都是在固定的可信任范围之内，认证可以简单的通过基于角色的授权和访问控制机制来实现。与传统的网络环境相比，网格具有分布与共享，自相似性与异构性，自治性与管理的多重性以及动态性等特点，网格协同计算环境中多个用户之间的信任关系变得十分复杂[6]，传统的安全模型很难适合于网格环境下多用户协同计算，因为：

(1) 无法表达任务约束，即网格协同计算环境下的安全协议必须在相应协同计算任务环境内执行，而协同计算任务形成的用户之间的协同关系会对安全协议的功能产生影响。

(2) 主要针对静态授权环境下的安全需求，无法适应网格协同计算任务执行过程中形成的用户权限的动态管理环境。

(3) 安全目标仅限于对应性和保密性安全需求，无法满足复杂网格环境的安全需求。

在网格计算中最能体现网格的特点，同时安全需求最为复杂的是“多用户参与的协同计算”。这种计算方式是指多个接入网格的用户之间需要协作共同完成某项计算任务。

针对这一特点，本文通过对网格环境下的“多用户参与的协同计算”中安全协议的需求分析和验证问题进行研究，建立起适用于网格协同计算环境的安全需求分析模型，同时提出了一种安全协议形式化分析方法。其主要包括以下内容：

(1) 提出了基于角色和任务的网格协同计算协同关系描述模型 TRBCR，而且在其基础上构建网格协同计算安全需求分析模型 TRBCC，实现了网格环境下多用户协同计算安全需求的形式化描述，从而把网格协同计算环境下的不同安全需求统一在一个理论体系中。

(2) 通过对传统 Strand Space 理论进行扩展，提出了一种基于虚拟组织的网格安全协议形式化验证方法，实现了网格协同计算环境下安全协议的分析与证明。

本文全文共分为五个部分，具体内容安排如下：

第一部分介绍了当前网格的研究现状；第二部分介绍了网格中的安全机制；第三部分提出了基于角色和任务的网格协同计算安全需求分析模型 TRBCC；第四部分提出了一种基于虚拟组织的网格安全协议形式化验证方法；第五部分通过实例研究对所提出安全模型和形式化验证方法进行验证。

1 网格基础

网格是借鉴电力网(Electric Power Grid)的概念提出来的。网格通过整合分布在局域网或广域网的资源，使之成为一个巨大的虚拟计算机系统。其目的是在大量个体、机构、组织之间利用安全、协同式的资源共享，创建一个动态虚拟组织[7](Virtual Organization，VO)。基于这种动态虚拟组织的网格协同计算不仅跨地域，而且延伸到不同组织中的异构软硬件平台，能够为每一个连接到网格的用户提供无限的计算能力、沟通协作能力及信息获得能力。

在网格协同计算环境下，通过采用基于虚拟组织的分布式管理模式，使得作业实体从资源控制、任务调度和管理的复杂工作中解脱出来。为了获得充分而必需的资源，各个VO可以使用标准的、开放的、通用的接口进行信息交互，并根据这些信息来协调各自的资源使用策略，从而避免系统的盲目查找和不合理的远程调用现象，以此大大提高了网格计算的智能性。在地域上分布的异构网格计算环境下，计算任务能自主地从某一计算节点迁移到另一计算节点，并可与其它VO的资源进行交互以实现作业和资源管理的自适应。

可以从以下三方面理解网格概念：

(1) 网格的目标是资源共享和协作。网格的这种概念可以清晰地指导对行业中各个部门的资源进行整体上的统一规划、部署、整合和共享。

(2) 网格是一种技术。为了达到多种类型的资源共享和协作，网格必须解决多个层面的资源共享和协作技术，制定相应的标准，从而将 Internet 从通讯和信息交互的平台提升到资源共享的平台。

(3) 网格是基础设施。网格通过集成各类计算资源，形成一种“无缝和集成的协同计算环境”，从而使得用户可以简便地获取提供的各种服务。

1.1 网格的特点

建立在现有的Internet和分布计算技术之上的网格，又被称为第三代的Internet，其基本特点有：

(1) 分布与共享：网格中的资源是分布的，分布的网格一般涉及的资源类型复杂，规模较大，跨越的地理范围广；同时，网格资源却是可以共享的，即网格上的任何资源都可以提供给网格上的任何使用者。分布是网格硬件在物理上的特征，而共享则是网格软件在逻辑上的特征。

(2) 动态性与多样性：网格资源是动态变化的。一开始网格的规模往往不是特别大，

但是网格能够对其自身进行多种形式的扩展。网格中的资源可以跨越地理分布的多个管理域，同时构成网格资源的计算机在体系结构、操作系统及应用软件等多个层次上可以具有不同的结构。

(3) 自相似性：相对于网格具有的动态性和多样性特点，网格提供了统一的资源调用方式，从而实现了异构资源的共享，并且在很大程度上提高了资源的有效利用。

(4) 自治性与管理的多重性：网格资源通常属于不同的机构或组织，因此网格资源的拥有者对该资源拥有自主的管理能力；同时，网格资源必须接受网格的统一管理。网格通过各个机构或组织共同参与虚拟组织来解决多级管理域的问题。

1.2 网格的发展现状

网格的发展经历了三个阶段：第一阶段是萌芽阶段，开始于 90 年代早期，研究内容是关于千兆网试验床以及一些元计算方面的工作；第二阶段是发展阶段，时间大概从 90 年代中期到晚期，这个阶段出现了一些比较重要的开创性和奠基性的研究项目，如 I-WAY，Globus，Legion 等；当前时期则是网格的迅速发展阶段，这一阶段关于网格的开发和应用项目大量出现，使得网格计算不再仅仅局限于科学研究，而是在军事和社会经济等更广泛的领域得到了推广和应用。目前有美国的 Globus、Legion、Condor、IPG，欧洲的 CERN DataGrid、UNICORE、MOL，澳大利亚的 Nimrod/G、EcoGrid，日本的 Ninf、Bricks，中国的国家网格、上海网格等网格研究项目。

当前网格应用主要分为以下四个方面：

(1) 分布式超级计算，将分布在不同地点的超级计算机用高速网络连接起来，并用网格中间件“粘合”起来，形成比单台超级计算机强大得多的计算平台，如用于大规模军事仿真的 SF Express。

(2) 数据密集型计算，侧重于数据的存储、传输和处理，其基本的思想是把海量数据分散到全球的计算机上进行处理，并实现结果的共享。如用于海量数据处理的 DataGrid。

(3) 分布式仪器系统，通过网格管理分布在各地的贵重仪器系统，提供远程访问和控制仪器设备的手段，提高仪器的利用率并方便用户的使用，如 X 射线设备的科学门户 Xport。

(4) 远程沉浸，是一种特殊的网络化虚拟现实环境，参与者通过网络聚在同一个虚拟空间里。从实现上看，虽然没有网格的支持，照样可以开发远程沉浸应用，但有了网格的支持，实现这类应用会简单得多，如 CAVE 虚拟现实环境。

1.3 网格体系结构

1.3.1 五层沙漏模型

五层沙漏模型[8]用于描述网格中的协议分层结构，这里的协议是指为了实现特定的操作而定义的分布式系统元素之间交互的方式以及交互信息的结构。强调“协议”的重要性是五层沙漏模型的显著特点。

类似于网络中的OSI协议分层结构，五层沙漏模型将整个网格环境中的各种协议分为五层，依次是：构造层(Fabric)，连接层(Connectivity)，资源层(Resource)、汇聚层(Collective)和应用层(Application)。在五层沙漏模型中，资源层和连接层共同组成了沙漏的瓶颈，是该模型的核心协议部分，起到了承上启下的作用。

构造层的基本功能是通过提供对局部资源进行控制的工具和接口，实现对所控制的共享资源的局部管辖与调度。构造层的资源非常广泛，可以是计算、存储、网络、数据和目录等。显然，构造层的实现依赖于特定的资源。

连接层的基本功能是定义了网格环境中的网络事务所需的通信和验证协议。其中，通信协议实现了构造层资源之间的信息交互，验证协议则是在通信协议的基础上，为用户和资源的识别提供加密的安全机制。

资源层的基本功能是通过定义单个资源的管理和操作的标准接口，实现资源的共享。资源层两个主要的协议是信息协议和管理协议。信息协议用来获得资源的结构和状态信息，管理协议用来协商对共享资源的访问。

资源层和连接层形成了沙漏模型的瓶颈部分，因此要求其包括的协议集合要小，而且要尽量标准化。这些协议要能够抓住涵盖不同资源类型的基本共享机制，而且又能足够支持各种类型的高层协议。

汇聚层建立在资源层之上，其主要功能是实现资源之间的协同。汇聚层的协议跨越了从通用到高端特殊应用领域的需求，其可以作为永久的服务来进行实现。

应用层的功能则是通过提供的相关工具来实现网格的应用和开发。

1.3.2 开放网格服务体系结构

开放网格服务结构OGSA(Open Grid Service Architecture)[9]是Globus小组和IBM于2002年初提出的一个面向服务的网格结构，是Web Service和Grid技术融合的产物。

(1) 基本思想

OGSA是以服务为中心的。OGSA在原有永久无状态Web服务的基础上，提出了临时网格服务的概念，用于解决网格资源的发现、创建、访问以及生命周期管理等有关的问题。

网格服务调用流程如图1.1所示。Factory服务发布到注册中心后，网格用户就可以查询到自己需要的Factory服务。用户向Factory服务发送服务创建请求，Factory服务创建网格服务实例，并将所创建的网格服务实例的网格服务句柄（Grid Service Handle，GSH）返回给用户。用户获得GSH后，继续通过句柄映射服务（HandleMap）中得到网格服务引用(Grid Service Reference, GSR)，并按照GSR的服务描述进行网格服务调用。

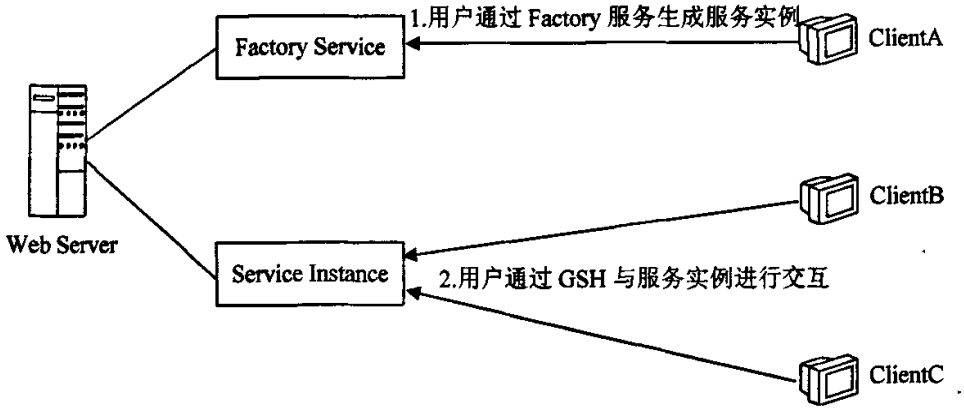


图 1.1 OGSA 服务调用流程示意图

Fig. 1.1 Service calling process in OGSA

(2) 网格服务框架

OGSA涉及以下方面内容：

① 网格服务资源

OGSA采用面向服务的体系结构，一个网格服务实例维护一个服务数据元素的集合。OGSA通过由WSDL定义的网格服务接口进行网格服务属性定义，并且隐藏了网格服务的具体实现，从而实现了资源的虚拟化和服务的无缝集成，增加了虚拟组织的灵活性。其中，网格服务质量、认证、授权等相关协议是通过与网格服务属性进行绑定来实现的。

② 网格服务的命名和绑定

一个网格服务实例被创建时都被分配一个唯一的GSH，GSH在网格服务的生存期内是不变的和唯一的。网格服务的其他信息保存在网格服务引用（GSR）中。与GSH不同的

是，GSR中的信息在网格服务的生存期内是可以发生变化的，如版本信息和协议绑定信息。HandleMap服务用于从GSH中解析出GSR，这个服务维护最新的GSH到GSR的映射，不会返回已经过时的GSR。

③ 创建临时服务

OGSA定义了一个接口来创建新的网格服务，实现这一接口的服务就是Factory。Factory服务从用户接收创建网格服务请求，并在成功创建网格服务之后返回该网格服务实例的GSH和初始的GSR。在OGSA中，Factory服务是分层次的，其上层的服务可以将工作委托给下层的服务。

④ 网格服务的生命周期管理

OGSA通过软状态来实现网格服务的生命周期管理。工厂在创建网格服务时，通过与用户进行协商，赋予网格服务一个初始生存期。创建网格服务后，用户可以通过SetTerminationTime操作来设置生存期，通过GetTerminationTime操作来查询生存期。在所设定的生存期到期后，网格服务提供者的运行环境或者网格服务本身可以决定销毁服务以释放资源。

⑤ 网格服务的注册和发现

网格服务的注册和发现主要包含二个方面：

注册管理接口，允许网格服务实例周期性地注册其 GSH 到注册中心；

服务发现接口，允许客户查询网格服务实例的信息，从而可以发现满足相关查询条件的网格服务实例集合。

⑥ 网格服务状态管理

网格服务的状态在生存期过程中会发生改变，这就需要对网格服务状态进行管理。OGSA通过异步的事件消息来表示网格服务状态的改变。OGSA定义了下列网格服务接口来实现事件消息的访问：

NotificationSource：支持用户向网格服务提供者订阅事件消息。

NotificationSink：支持网格服务提供者向用户发布的事件消息。

用户首先通过 NotificationSource 服务接口向网格服务提供者订阅其所关注的网格服务状态变化；同时，当网格服务状态变化时，网格服务提供者负责通过 NotificationSink 服务接口通知所有订阅的客户。

(3) 网格服务接口与功能机制

开放网格服务基础设施OGSI(Open Grid Service Infrastructure)[10]是构建OGSA的基础设施，其通过对Web服务进行扩展，定义了网格服务的规范。

OGSI提出了服务数据(Service Data)的概念。用户通过网格服务接口对网格服务实例的服务数据进行访问，包括关于服务数据的增加、删除、订阅等相关操作。服务实例在运行过程中，可以动态增加和删除其服务数据。

OGSI通过WSDL定义网格服务，由于原有WSDL所定义的元素不足以描述网格服务的特征，OGSI对WSDL进行了扩展，其中包括对WSDL中的portType定义了新的模式，以及提出支持portType的继承模型，在该继承模型中，继承属性会影响继承链上所有的服务数据的定义。

OGSI定义了用于网格服务管理的核心服务接口，根据其功能可以分为三类，如下所示：

① 支持网格服务数据访问的接口。其中，Factory 用于创建网格服务实例；Registry 用于注册一个 GSH；GridService 用于检索已注册的 GSH 的消息；HandleMap 用于把 GSH 解析为有效的 GSR。

② 支持网格服务状态管理的接口。其中，NotificationSource 用于用户向网格服务提供者订阅事件消息；NotificationSink 用于网格服务提供者向用户发布事件消息。

③ 支持网格服务分组的接口。其中，ServiceGroup 是代表网格服务组的抽象接口；ServiceGroupRegistration 接口继承自 ServiceGroup 接口，用于管理网格服务组的操作，包括向一个网格服务组中增加和删除成员网格服务；ServiceGroupEntry 接口则是由 ServiceGroupRegistration 接口创建的，用于定位网格服务组中的成员网格服务。

1.3.3 WSRF

针对OGSI的一些缺点，全球网格论坛GGF的OGSI工作组(OGSI-WG)于2004年提出了Web服务资源框架(Web Service Resource Framework, WSRF)[11]。

(1) WS_Resource

在Web Service永久无状态服务的基础上，WSRF引入了状态资源的概念，并且将服务与状态资源区分开来，一个服务可以对应多个状态资源，一个状态资源也可以对应多个服务。服务与状态资源的组合构成了WS_Resource。

WSRF引入状态资源的动机是：虽然服务在交互过程中并不维护资源状态信息，但是整个交互过程则需要资源状态信息，也就是说，资源状态通过服务交互得以持久化，并且作为服务交互的结果而保存。

另外，在服务交互过程中得到处理的不是状态资源本身，而是对状态资源的引用，这个引用是状态资源的惟一标识，称为Web服务端点引用，主要包括如下属性：

① [address]：标识状态资源所在的网络地址；

② [reference properties]：标识得到处理的状态资源；

③ [policy]：包含服务交互所需的策略信息。

针对WS_Resource，WSRF进一步制订了相应规范，其中某些规范还在制订过程中。

(2) WSRF 规范

WSRF规范按照功能分为六部分，其中每个部分都可以独立的发展，也可以通过多种方式进行组合。具体描述如下：

① WS_ResourceProperties 规范用于定义 WS_Resource 的类型以及其对应的 Web 服务端点引用如何与原有的 Web 服务端点描述相关联，同时也定义了 WS_Resource 属性的访问方式。

② WS_ResourceLifetime 规范用于 WS_Resource 的生命周期管理，包括 WS_Resource 的创建，标识和销毁。WS_Resource 是由 Web 服务资源工厂(WS_Resource Factory)来创建的。

③ WS_RenewableReference 规范定义了一种机制，通过这种机制可以对已经无效的 Web 服务端点引用进行更新，从而实现持久、稳定的 WS_Resource 的引用。

④ WS_ServiceGroup 规范定义了一种机制，通过这种机制，Web 服务和 Web 服务资源可以为了某个领域的特定目的而组合在一起。

⑤ WS_BaseFaults 规范定义了基本故障的 XML 模式类型以及这种故障类型的使用规则，从而实现对来自于不同 Web 服务的故障进行一致的处理。

⑥ WS_Notification 规范将企业级发布和订阅消息模式集成到 Web 服务中。其提供了直接发布和代理发布两种模式，在代理发布模式中，代理将消息的发布者和订阅者分离开来。

2 网格安全机制

在计算机通信网络中，主要的安全保护措施被称作安全服务。根据 ISO7498-2，安全服务包括数据的机密性，鉴别，数据的完整性，不可抵赖性和访问控制。作为一种新出现的重要的基础性设施，网格对于运行的可靠性提出了更高的要求。

2.1 网格安全机制

网格环境下的标准安全功能包括认证、访问控制、完整性、隐私权保护和抗抵赖性。相对于网格环境下的安全体系结构，需要满足以下基本的限制条件：

(1) 单一登陆点：用户只需要在开始计算是进行一次认证；在获得资源、释放资源、内部通信时无需对用户再次进行认证。

(2) 信用凭证保护：用户的凭证(密码，密钥等)必须受到保护。

(3) 局部安全方案的互用性：当安全解决方案能提供域间访问控制机制时，对局部资源的访问应由本地安全机制决定。但是，改变局部资源来适应域间访问是不现实的，这要求有一个或多个实体作为局部资源的远程客户/用户代理。

在所有满足这些要求的安全体系结构中，认证和授权构成了整个网格安全的基础。

目前 Globus[12]是应用最广泛的网格项目，由于它的开源性，现在的许多网格项目（如欧洲数据网格，分布式异构计算环境 Cactus 等）都是基于它发展起来的。Globus 所采用的安全机制是 GSI(GridSecurity Infrastructure)[13,14,15]。GSI 主要解决了认证和消息保护问题，CAS(Community Authorization Service)[16,17]基于 GSI, 并通过在每个 VO(Virtual Organization)中引入一个 CAS 服务器，很好地解决了授权问题。

2.2 网格安全基础设施 GSI

GSI是基于公钥加密、X.509证书和SSL通信协议的一种安全机制，主要解决了VO中的认证和消息保护问题。在早期的一些工作中，GSI使用私钥和X.509证书作为用户向网格资源认证的基础。GSI 的主要安全目标包括：

(1) 支持网格计算环境中的安全通信；

(2) 支持跨虚拟组织的安全；

(3) 支持网络计算环境中用户的单点登陆，包括跨多个资源和地点信任委托和信任转移等。

现在，GSI 使用一种称为代理证书(Proxy Credential)的临时凭证。通过代理证书，GSI 用户在访问分布在多个站点上的资源时就不必重复认证，亦可实现单点登录。另外，

通过代理证书 GSI 用户还能够将用户权限委托(Delegation)给远程进程，其具体描述如下：

(1) 单点登录

在一个网格计算环境中,用户可能需要同时访问多个站点上的资源,这样就有可能要求用户进行多次认证,从而要求用户多次键入其私钥加密密码。这无论从方便性还是安全性的角度都是不可取的。

GSI使用代理证书解决了这个问题。用户生成一对新的公私钥对,并用其用户证书签发生成一个新证书。新证书包含了用户的身份和新生成的公钥,并被略作修改以指示这是一个代理证书。代理证书是由用户签发的一个短暂证书。代理凭证包含了新生成的私钥、代理证书和用户证书。用户在使用代理证书认证时，必须向验证方同时提供用户证书和代理证书。验证方首先检查用户证书的有效性，然后检查代理证书是否为用户所签发，最后判断用户是否拥有代理证书的私钥。若条件都满足,用户认证成功。

代理证书的私钥可以以明文的形式保存在本地存储系统中,并禁止其他用户对私钥的读取。这是因为代理证书是一个短期的证书,所以代理证书的私钥不必像用户私钥那样加密保存。代理证书一旦创建,在证书过期前，VO用户可以用它同资源提供者进行相互认证,而无须再键入原有用户证书的私钥密码，从而提高了系统的安全性。

代理证书除了包含用户身份之外,还可以在其扩展项中承载一些策略信息来限制其使用,包含有这样的策略信息的代理证书叫做受限代理(Restricted Proxy)。在后面部分我们会看到受限代理在CAS中的使用。

(2) 委托

在分布式应用中，用户提交一个作业到某个远程站点A上去执行，这个作业可能需要访问用户保存在另一个远程站点B上的文件。这样,用户在站点A上的作业需要能够以用户的身份访问保存在站点B上的文件。在这种情况下，GSI则允许用户委托一个代理证书给其在站点A上的作业,从而用户在站点A上的远程作业能够使用用户委托的代理证书通过站点B的相关系统认证。

(3) 授权

关于授权的一种解决方案是，将属于不同独立组织的资源和/或人员组织起来创建一个VO，继而在VO中基于计算任务采用动态的策略来监控到底谁可以使用哪些资源,以及出于什么目的使用这些资源。

VO中的资源共享是通过服务实现的，控制用户对资源服务的访问权限是通过一个映射文件来实现的。映射文件由一系列用户DN(区别名)到本地账号(如一个本地的UNIX用户账号)的映射项组成。用户认证成功后,资源提供者从用户的代理证书中提取

用户DN，然后资源提供者根据用户请求的服务从相应的映射文件中查看是否有该DN的映射项。如果存在，则说明用户有权限访问其请求的服务，资源提供者将以用户DN对应的本地账号运行被请求的服务。这样一来,用户请求的这个资源服务的行为将受到本地账号的约束。换句话说，GSI是通过映射机制将对用户的访问控制转变为资源提供者对本地账号的访问控制，如文件访问控制、CPU限制等。

GSI采用资源提供者和资源消费者之间直接的信任关系(映射文件中是否有用户DN的映射项)来描述授权策略，所以对大规模的VO，这样会有扩展性、灵活性、描述性以及缺少策略层次性的问题。例如，当一个用户加入VO时,可能需要同所有资源提供者交互以建立信任关系；当一个资源提供者加入VO时，也可能需要同所有VO用户交互以建立信任关系。CAS服务器是VO为解决这些问题引入的一个可信第三方，负责管理监控用户对VO中资源的访问。用户要访问VO中的资源时，首先要同CAS服务器打交道，CAS服务器根据用户的请求和其在VO中的角色授予一定的权限给用户。CAS服务器通过为用户签发代理证书并且把用户的相关权限写入到代理证书的策略描述中，实现了VO对用户的授权，相应地，CAS服务器为用户签发的代理证书称为受限代理。用户继而可以将代理证书出示给资源提供者获取相关资源的访问权限。用户最终得到的有效的权限是资源提供者授予CAS服务器的权限集合与CAS服务器授予用户的权限集合的交集。

资源提供者和资源消费者通过CAS服务器建立信任关系，使得加入VO的实体只需要同CAS服务器打交道，因此扩展VO变得相当容易。资源提供者提供了哪些共享资源，这些资源用于什么目的等都可直接在CAS服务器中描述(CAS提供了策略描述语言)，这又使得我们可以非常方便灵活地反应VO策略的变化。最后，我们还可以通过建立另一个CAS服务器，负责管理子VO的策略来解决策略层次性问题。

2.3 Globus Toolkit 4 的安全实现

Globus ToolKit 是 Globus 网格计算项目在多种平台上运行的网格计算工具包软件。Globus 项目提出通过建立网格安全基础设施 GSI 来保障网络计算环境的安全。GSI 已经在 Globus ToolKit 4（GT4） 中得到了实现。

在当前的 Globus 环境中，GSI 实现的安全策略能够满足以下条件：

(1) 用户代理和资源代理之间的所有连接需要安全鉴定。

(2) 所有安全鉴定是相互的。

(3) 一个用户代理在一个限定时间内可代表一个用户与一个资源代理进行交互。

(4) 具有相同资源的资源代理间相互信任。

(5) 一个进程管理者和它创建的进程假设在一个单一的信任域中执行。

(6) 由相同的用户代理/资源代理创建的所有进程之间彼此信任。

(7) 资源的存取控制权限由其所处环境中的本地安全策略确定，并通过本地安全机制来实现。

(8) 所有的 Globus 特定主体可通过映射到一个资源特定主体(如用户 ID)来实现本地存取机制。

2.3.1 GT4 安全机制

(1) 用户和服务证书。GSI 的安全机制基于公钥加密，采用 X.509 认证和 SSL 通信协议。GT4 使用与 GT2 相同的用户和服务身份证书，支持代理证书、支持授权和单一登陆。参与其中的实体通过其持有的证书，以及事前建立的软件与过程，实现身份的表示和确认。

(2) 认证中心。GT4 网格认证中心（Certification Authority，CA）可以通过 Globus 小组发布的 Simple CA 包生成 CA，从而可以在 Globus 网格中发出证书。

(3) 资源授权。GT4 的授权基于简单的访问控制列表，这个列表位于明文文件 Gridmap 中，Gridmap 文件与服务和 factory 相关联，用于限制谁可以访问所提供的功能。在任务提交的过程中，GT4 用这个 Gridmap 文件将用户映射为远程资源上的用户 ID，并实现了对 factory 和服务的访问控制策略。

(4) 应用程序接口。仍然是 Generic Security Service API，GSSAPI 定义提供了通用的安全服务，支持各种安全机制和技术，还支持应用程序在源码级的可移植性。GSSAPI 主要面向主体之间的安全鉴别和安全通信操作，提供的功能主要包括：获得证书、执行安全鉴别、签署消息和加密消息。

GT4 为了支持 OGSA，对网格安全机制作了改进。包括：

(1) 增加了 Web Service 的安全技术，包括 Web Services SecureConversation 协议、XML-Encryption、XML-signature。在 GT4 中，相互认证完全通过标准 SOAP 层上实现，从而允许使用任何底层传输实现 SOAP 协议。SOAP 层的安全基于 Web Service 安全机制、XML 加密及签名标准。

(2) 对资源安全模型进行了改善。采取直接接收来自于网络的服务，进程本身不再拥有特殊的本地全县，使用两个特殊的 setuid 进程完成需要权限的动作，从而加强服务的安全性。

(3) 从信任模型中剔除了面向网络的服务。GSI 在 GT4 中的具体实现通过 Java GSSAPI 来实现，它仍然继续支持相互认证、安全委托、信息保护以及授权。GT4 定义的实现上述安全机制的协议都是基于 SSL，为了在 SSL 上实现 GSI，允许信任委托和

代理签名在 SSL 握手后执行，对 HTTPS 协议加以改进，改进后的 HTTPG 协议可以实现安全委托，并能在 SSL 密钥协商后实现代理签名。

2.3.2 GT4 的计算任务提交与执行过程描述

下面介绍在GT4(Globus ToolKit)中计算任务的提交与执行过程描述。

在图 2.1 所示网格计算过程 P 中，用户 User 属于虚拟组织 VO_1，RN_1，RN_2，RN_3 是属于 VO_1 的资源，RN_4，RN_5 是属于虚拟组织 VO_2 的资源，GN_1，GN_2 分别对应虚拟组织 VO_1，VO_2 中的 gatekeeper。

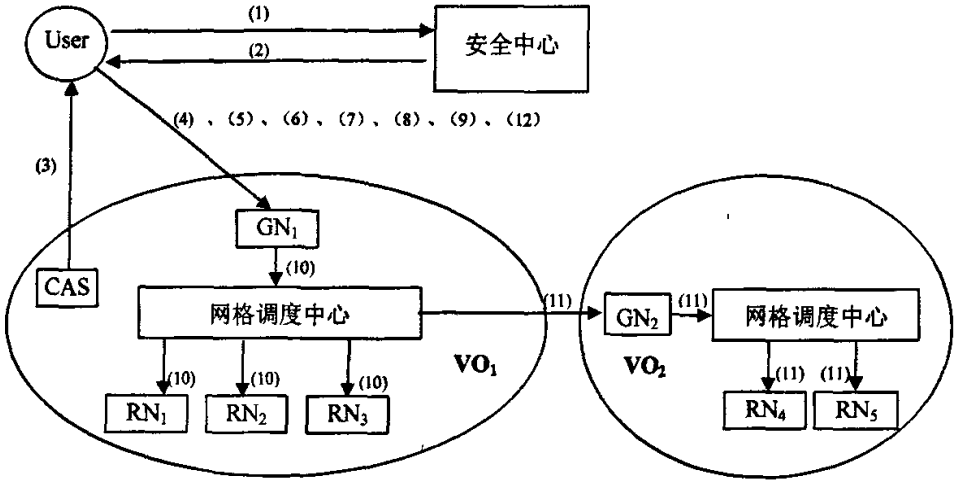


图 2.1 GSI 计算任务提交与执行过程示意图

Fig. 2.1 Submit and execute process of computing tasks in GSI

可对图2.1中的计算任务的提交与执行过程作如下说明：

(1) 在进行 Globus 任务提交与执行之前，用户与服务节点需要获得安全证书。即可以使用命令行命令 grid-cert-request 或对应得安全函数创建用于安全鉴别的公钥、私钥和未签署的安全证书。然后通过 email 或其他安全途径把它提交给安全中心。

(2) 安全认证中心收取签署安全认证证书的请求后，会对用户或服务节点进行考察，考察合格后，把签署过的安全认证证书返回给请求方。

(3) 用户在提交任务前，可通过命令行命令 grid-proxy-init 或对应的安全函数向 CAS 服务器请求一个受限代理。受限代理由 CAS 服务器签署，由给定的有效期。

(4) 用户代理签署计算任务的描述，交给资源的 MMJFS(Master Managed Job Factory Service，主控受管理作业工厂服务)。

(5) MMJFS 属于一个几乎没有任何权限的用户，验证任务请求和受限代理证书，确定用户标识和需要提供服务的本地用户。

(6) 如果本地用户没有 LMJFS(Local Managed Job Factory Service)，MMJFS 将向 setuid starter 申请启动新的 LMJFS，setuid starter 具有 root 权限，调度用户的 LMJFS。

(7) LMJFS 启动后，访问 GRIM(Grid Resource Identity Mapper, 另一个 setuid 进程)，访问本地主机的证书并为 LMJFS 用户产生代理证书。

(8) MMJFS 把原始的用户签署的计算任务描述给 LMJFS，LMJFS 验收并授权。然后启动一个 MJS 的应用返回给请求用户。

(9) 用户代理和 MJS 交互鉴别(MJS 的证书从 GRIM 申请获得)。

(10) 用户通过 MJS 在相应资源上执行计算任务。

(11) 如果计算任务在执行过程中需要访问远程资源，也必须在任务进程与远端文件服务资源代理之间进行相互安全鉴别。通过安全鉴别后，还要进行授权、本地 id 映射后，任务进程才可以进行远端数据或文件访问。

(12) 当任务执行完后，用户可以通过 grid-proxy-destroy 或对应的函数撤销用户代理。

3 基于角色与任务的网格安全需求分析新模型 TRBCC

本章论述基于角色与任务的网格安全需求分析模型 TRBCC。其基本思想是：通过建立 TRBCR 模型对网格动态授权环境中的协同关系进行描述，并且进一步在 TRBCR 模型基础上提出了一种形式化安全需求描述语言 TRBCL，最终实现了网格环境中多用户协同计算安全需求的形式化描述。TRBCC 模型又简称作安全空间。

3.1 现有安全需求分析模型的解析

安全模型的设计主要依靠安全建模技术来实现，即通过对系统安全需求的分析，建立系统的静态和动态安全需求的形式化描述。对系统安全需求进行形式化描述有助于理解协议设计目的。目前人们在安全协议中采用形式化方法进行研究时，主要工作集中在了协议安全性的验证方面。而在协议安全需求的形式化描述方面虽然已经做了一些工作，但此领域仍然很少涉及。

目前,在该领域的研究大致有以下两个方向:

(1) 建立通用的认证协议安全需求

Diffie、van Oorschot 和 Wiener[1]对认证协议安全需求进行了非形式化描述。他们认为，认证协议的会话密钥应该保持秘密，协议应该匹配运行。其中，协议匹配运行是指如果协议参与者 A 与 B 运行协议，那么 A 接收来自 B 的消息数据和 B 发送给 A 的消息数据要求匹配。Bellare，Rogaway[2]和 Syverson[3]分别对协议的匹配运行安全需求进行了形式化描述，并在此基础上提出各自的安全需求分析模型。目前，得到广泛认可的是 Woo 和 Lam[4]提出的认证协议的安全需求定义。Woo 和 Lam 认为认证协议的基本功能是认证和密钥分配,那么相应地，就具备两个基本安全属性：对应性和秘密性。对应性是指对于一个协议主体的事件发生，必定有相对的另一主体的相关事件发生，两个事件之间有严格的时间先后关系。对应性和协议的认证功能相对应。对应性和匹配运行的描述非常相似，却比匹配运行的应用范围更广泛，因为对应性中的关联事件并不仅指同一个消息数据的发送和接收。

(2) 针对不同类型协议,建立不同的安全需求

Syverson 和 Meadows[5]在为 NRL 协议分析器所设计的语言中首次采用了针对不同类型的协议，建立不同的安全需求的方式。其安全需求定义和 Woo-Lam 模型中的对应性概念在形式上有一些相似，都基于事件的发生次序。但是，和 Woo-Lam 模型中对应性安全需求可应用于所有协议不同，语言使用者可针对不同协议定义不同安全需求。它

们已经给出了针对各种消息认证协议和可重复认证密钥分配协议的安全需求形式化描述方法。

对应于网格环境的复杂性，网格安全问题错综复杂，为了满足各种安全需求，必须根据较为完善的安全模型来设计针对不同安全目标的安全协议。

对于传统网络环境下的协同计算，由于参与者和使用的资源都是在固定的可信任范围之内，认证可以简单的通过基于角色的授权和访问控制机制来实现。与传统的网络环境相比，网格具有分布与共享，自相似性与异构性，自治性与管理的多重性以及动态性等特点，网格协同计算环境中多个用户之间的信任关系变得十分复杂[6]，传统的安全模型很难适合于网格环境下多用户协同计算，因为：

(1) 无法表达任务约束，即网格协同计算环境下的安全协议必须在相应协同计算任务环境内执行，而协同计算任务形成的用户之间的协同关系会对安全协议的功能产生影响。

(2) 主要针对静态授权环境下的安全需求，无法适应网格协同计算任务执行过程中形成的用户权限的动态管理环境。

(3) 安全目标仅限于对应性和保密性安全需求，无法满足复杂网格环境的安全需求。

在网格计算中最能体现网格的特点，同时安全需求最为复杂的是“多用户参与的协同计算”。这种计算方式是指多个接入网格的用户之间需要协作共同完成某项计算任务。针对这一特点，本章通过对网格环境下的“多用户参与的协同计算”中安全协议的需求分析和验证问题进行研究，建立起一种适用于网格协同计算环境的安全需求分析模型。

3.2 网格计算多用户协同关系描述模型（TRBCR）

TRBCR 是一种基于角色与任务的网格计算协同关系描述模型，其通过引入网格计算任务以及角色扮演者的概念，实现了对网格动态授权环境中协同关系的描述，通过在其基础上建立的网格安全需求的形式化描述语言 TRBCL，可以精确的把握网格系统的安全需求，从而可以有效的布置安全策略，设计安全协议。

在定义 TRBCR 模型之前，首先给出以下几个定义。

定义 1.(基本集合) 用户集 U，角色集 R，时间片集 T，消息数据集合 M。

其中，时间片用于表达网格计算任务相关的时间限制，消息数据则用于描述网格计算任务执行过程中，网格计算任务参与者之间的交互信息。

定义 2.(网格计算任务，Ct) 网格计算任务是用来表示网格环境下的多用户协同计算的逻辑单元，它包括一系列可区分的动作，可能与多个用户相关,也可能包括几个子任务。其

集合记为 *CT*。对于网格计算任务 Ct_1，Ct_2，如果 Ct_2 是 Ct_1 的子任务，则可表示为 $Ct_1 \angle Ct_2$。

定义 3.(任务状态，State) 任务状态是指网格计算任务在执行过程中可能经历的状态。令 $Ct \in CT$，则该网格计算任务的状态为 Ct[state]，在我们的模型中，有如下 4 种任务状态：

(1) 就绪状态(wait). 表示一个任务的运行条件已经满足,完成了运行前的准备工作。

(2) 运行状态(running). 表示一个任务正在运行。

(3) 正常终止状态(finished). 表示一个任务由于完成而终止。

(4) 异常终止状态（failed). 表示一个任务由于出现异常而终止。

定义 4.(角色扮演者，Actor) 角色扮演者是由一个角色被用户激活后产生的一个动态对象,是用户执行该角色的代理。可以用一个元组〈User，Role, Lifetime〉表示,其集合记为 *A*；其中:

(1) User 是 Actor 所代理的用户且 $User \in U$;

(2) Role 是所激活的角色且 $Role \in R$;

(3) Lifetime 为 Actor 的生存时间且 $Lifetime \in T$。

(4) Actor.User，Actor.Role，Actor.Lifetime 分别对应元组中相应元素。

在目前已有相关定义中，角色扮演者一般用来描述用户使用资源的权利[13]，在 TRBCR 模型中，则通过角色扮演者对网格计算任务参与者在网格动态授权环境中的相关协同关系进行描述。

定义 5.(角色管理者，Rm) 角色管理者负责相关角色的激活与授权。其集合可记为 *RM*。其中，角色的激活指将用户转化为相应角色扮演者，授权则指赋予角色扮演者参与相应网格计算任务的权利。

定义 6.(认证中心，Ca) 在网格计算环境中，认证中心负责颁发用户的公钥证书。其集合记为 *CA*。

定义 7.(证书，Cert) 认证中心颁发给用户的公钥证书，其集合记为 *Cer*。

在 TRBCR 模型中，通过把资源提供者与资源使用者统一角色化，并且进一步提出网格计算任务与角色扮演者的概念，实现了网格动态授权环境中协同关系的描述，从而为下一步的安全需求描述打好基础。图 3.1 是 TRBCR 模型示意图。

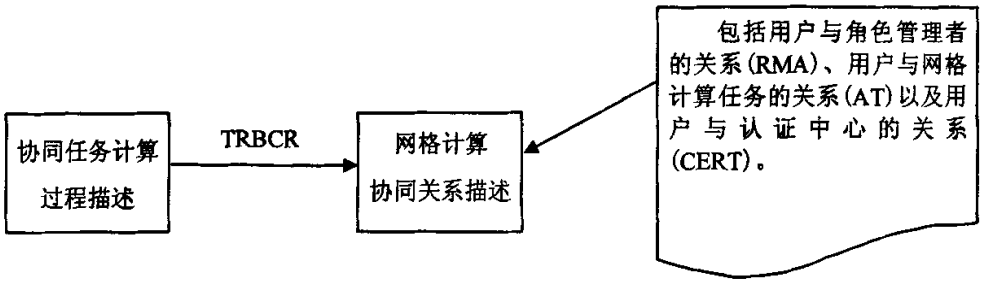


图 3.1 TRBCR 协同关系描述模型示意图

Fig. 3.1 TRBCR model

网格动态授权环境中协同关系包括用户与角色管理者的关系(RMA)、用户与网格计算任务的关系(AT)以及用户与认证中心的关系(CERT)。

在 TRBCR 模型中，用户一旦激活某个角色，相应地激活一个角色扮演者，角色与角色管理者(RMR)以及角色与任务(RT)之间的静态关系则会通过动态授权相应地传递到角色扮演者中，这样用户参与网格计算任务的过程就映射为角色扮演者参与网格计算任务的过程。 这个过程是一个动态过程。图 3.2 是 TRBCR 模型中协同关系动态描述示意图，其通过 RMR，RT 向 RMA，AT 关系的动态转化反映了网格动态授权环境中的协同关系。此外，用户与认证中心的关系会影响用户的激活与授权，从而进一步对网格动态授权环境的其他协同关系产生影响。

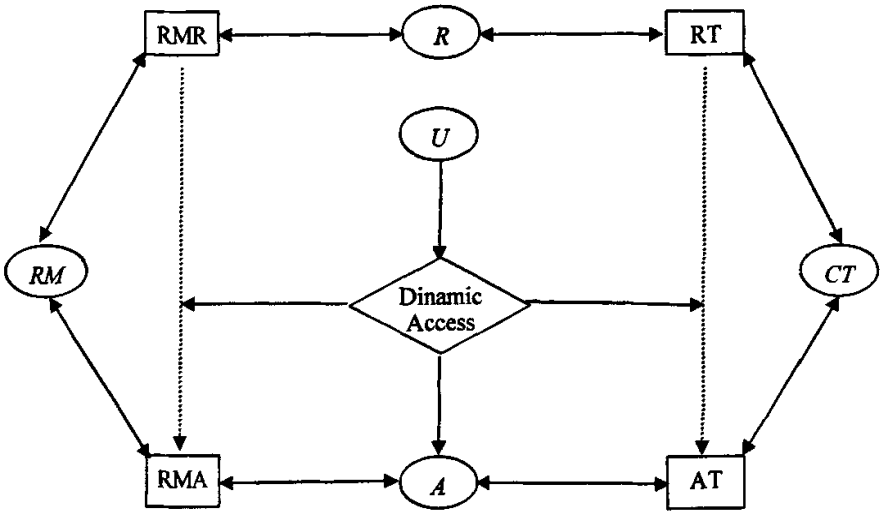


图 3.2 TRBCR 模型中协同关系动态描述示意图

Fig. 3.2 Dynamic description of coordinative relationship in TRBCR

定义 8.(TRBCR 模型) TRBAR = （U，R，T，A，CT，RM，CA，Cer），其中 U 是用户集合，R 是角色集合，T 是时间片集，CT 是网格计算任务集合，RM 是角色管理者集合，CA 是认证中心集合，Cer 是公钥证书集合。定义下列关系：

(1) $RT \subseteq R \times CT$ 是多对多角色到网格计算任务的指派关系。

(2) $AT \subseteq A \times CT$ 是多对多角色扮演者到网格计算任务的指派关系。

(3) $RMR \subseteq RM \times R$ 是一对多的角色管理者到角色之间的管理关系。

(4) $RMA \subseteq RM \times A$ 是一对多的角色管理者到角色扮演者之间的管理关系，用户激活角色时，角色管理者到角色之间的管理关系就映射为角色管理者到角色扮演者的管理关系。

(5) $Obliged\_rm: R \rightarrow RM$ 是多对一的角色到其角色管理者的映射。

(6) $Domain: U \rightarrow 2^{CA}$ 是一个信任域映射，它将每个用户映射到其信任的认证中心的集合。

(7) $CERT: CA \times U \rightarrow Cer$ 是一个公钥证书映射。

定义 9.(子空间) 对于 $TRBCR_1=(U_1, R_1, T_1, A_1, CT_1, RM_1, CA_1, M_1, Cer_1)$, $TRBCR_2=(U_2, R_2, T_2, A_2, CT_2, RM_2, CA_2, M_2, Cer_2)$, 如果 CT_2 是对应于 $Ct \in CT_1$ 所有子任务组成的网格计算任务集合，则 $TRBCR_2$ 所属安全空间 $TRBCC_2$ 称为 $TRBCR_1$ 所属安全空间 $TRBCC_1$ 的子空间。表示为 $TRBCC_2 \angle_{Ct} TRBCC_1$。

3.3 TRBCC 安全空间

TRBCC 安全空间是基于角色与任务的网格安全需求分析模型，实现了网格环境下多用户参与的协同计算安全需求的形式化描述。该模型由三部分组成：网格计算多用户协同关系描述模型 TRBCR、网格计算信息交互图和网格安全需求的形式化描述语言。TRBCC 模型建立过程如下：

根据网格计算任务描述首先建立起网格计算多用户协同关系描述模型 TRBCR，对在网格动态授权环境下的协同关系进行描述；然后在 TRBCR 模型基础上构建网格计算信息交互图，形成对网格计算任务参与者之间消息数据交互和处理过程的描述；并且进一步根据本文提出的一种网格安全需求的形式化描述语言，完成对网格动态授权环境下安全需求的形式化描述。图 3.3 说明了 TRBCC 安全空间各组成部分之间的关系。

其中，TRBCR 协同关系描述模型引入了角色扮演者和网格计算任务以及角色管理者的概念，使之能够准确的描述网格动态授权环境下的协同关系，通过在其基础上建立的网格安全需求的形式化描述语言 TRBCL，可以精确的把握网格系统的安全需求，从而可以有效的布置安全策略，设计安全协议。

3.3.1 网格计算信息交互图

网格计算信息交互图是一种基于 TRBCR 模型的赋图结构，用于描述网格计算任务参与者之间的消息数据交互和处理过程。

定义 10.(迹) 用形如$<<\varepsilon_1,\alpha_1>,<\varepsilon_2,\alpha_2>,\ldots<\varepsilon_n,\alpha_n>>$的有限序列表示一个事件序列，其中，$\varepsilon_i \in \{+,-\}$(+表示接收消息数据，-表示发送消息数据)，$\alpha_i \in M$ $(1<i<n)$，表示消息数据，称这样的一个事件序列为一个迹(trace)。

定义 11.(网格计算信息交互图) TRBCR 中的每个角色扮演者和网格计算任务对应一个迹，把迹中的每一个二元组表示为图的一个节点 n=<s,k>，其中 s 代表一个迹，k 表示 s 的第 k 个分量。term(n)表示节点 n 有符号的消息数据，uns_term(n)表示节点 n 没有符号的消息数据。如果 $term(n_1)=+\alpha$, $term(n_2)=-\alpha$(其中 $\alpha \in M$, $term(n)=+\alpha$ 表示节点 n 对应事件为发送消息数据 α，$term(n)=-\alpha$ 表示节点 n 对应事件为接受消息数据 α)，并且信息交互一方为角色扮演者 Actor，另一方为网格计算任务 Ct，同时满足<Actor.Role，Ct>∈

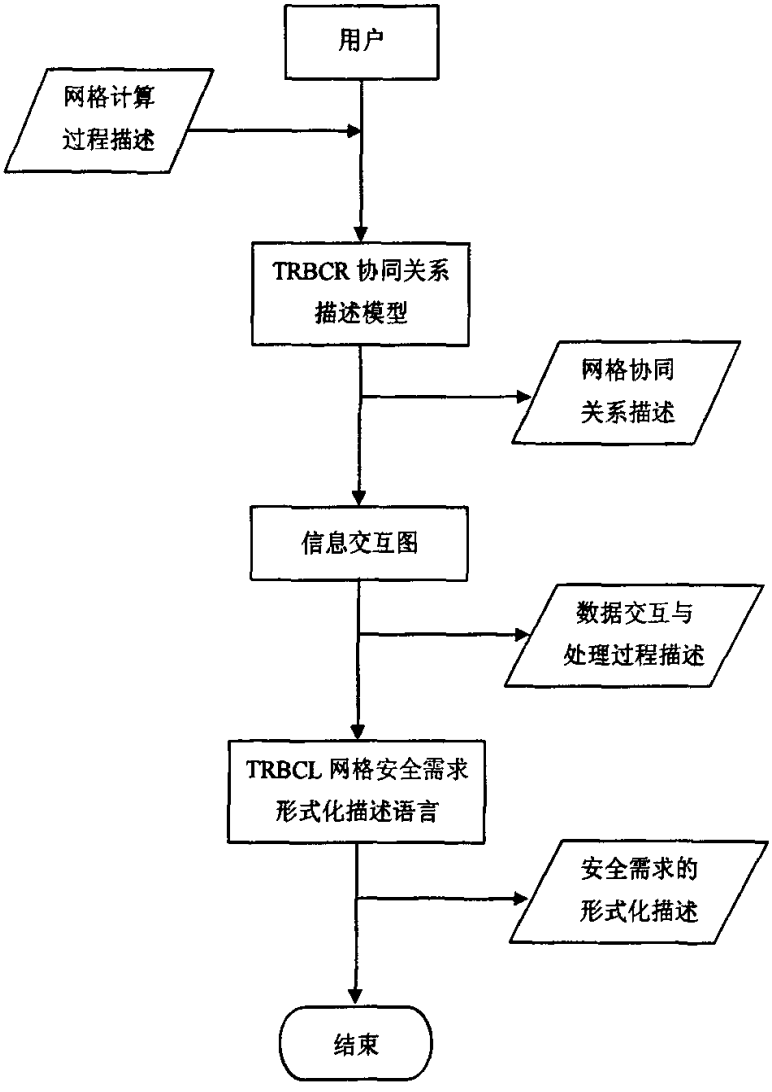


图 3.3 TRBCC 安全空间建立过程示意图

Fig. 3.3 Building process of TRBCC

RT,则记为 $n_1 \rightarrow n_2$；如果 n_1=<s,i>，n_2=<s,i+1>，则记为 $n_1 \Rightarrow n_2$。由此生成的有向图称为基于 TRBAC 的网格计算信息交互图。

定义 12.(知识集) 角色扮演者在网格计算任务执行过程中所拥有的数据集合称为知识集。表示为 A[actor] $\subseteq 2^M$。其中，在网格计算任务开始执行之前，角色扮演者有一些已知数据，当网格计算任务开始执行之后，角色扮演者的知识集会发生变化。

3.3.2 基于角色和任务的网格安全需求的形式化描述语言 TRBCL

基于角色和任务的网格安全需求的形式化描述语言 TRBCL 用于描述网格计算任务的安全需求，其格式表示为：

RANGE <TRBCC 安全空间>

GET A(<角色扮演者>):<条件表达式 1>

SECURITY REQUIREMENTS:<条件表达式 2>

其中，TRBCC 安全空间表示网格安全需求的适用范围；角色扮演者从属于当前 TRBCC 安全空间；条件表达式 1 定义关于角色扮演者的选择条件，默认条件为当前 TRBCC 安全空间中的所有角色扮演者；条件表达式 2 则通过所选角色扮演者对网格安全需求进行定义。

在 TRBCR 模型和信息交互图的基础上可以通过定义网格计算过程中的事件与基本安全需求的逻辑表示来实现网格安全需求的形式化描述。

定义 13.(事件) 事件用来描述网格计算过程中参与主体的行为，其集合表示为 *EVENT*，包括以下内容：

(1) Authen($Actor_1$，$Actor_2$，Cert)：Cert$\in$*Cer*，$Actor_1$，$Actor_2 \in A$，表示角色扮演者 $Actor_1$ 通过证书 Cert 对角色扮演者 $Actor_2$ 进行身份验证。

(2) Authen(Rm，Actor，Cert)：Cert$\in$*Cer*，Rm$\in$*RM*，User$\in U$，表示角色管理者 Rm 通过证书 Cert 对角色扮演者 $Actor_2$ 进行身份验证。

(3) Activate(Rm，Actor)：Actor$\in A$，Rm$\in$Obliged_rm (Actor.Role)，表示角色扮演者 Actor 被角色管理者 Rm 激活。

(4) Author(Rm，Actor，Cts，Lifetime)：Actor$\in A$，Rm=obliged_ rm(Actor)，Cts$\subseteq CT$，Lifetime$\in T$。指在网格计算过程中角色管理者 Rm 对角色扮演者 Actor 的一次授权。Cts 表示角色扮演者 Actor 经过授权所能够参与的网格计算任务集合，Lifetime 表示此次授权的持续时间。

(5) Revoke(Rm，Actor，Cts)：Rm=obliged_rm(Actor)，Actor$\in A$，Cts$\subseteq CT$。指在网格计算过程中角色管理者 Rm 终止角色扮演者 Actor 参与 Cts 中网格计算任务的权利。

(6) Send($Actor_1$，$Actor_2$，m)：$Actor_1$，$Actor_2 \in A$，$m \in M$，角色扮演者 $Actor_1$ 向角色扮演者 $Actor_2$ 发送消息数据 m。

(7) Receive($Actor_1$，$Actor_2$，m)：$Actor_1$，$Actor_2 \in A$，$m \in M$，角色扮演者 $Actor_1$ 接受到来自角色扮演者 $Actor_2$ 的消息数据 m。

定义 14.(基本安全需求) 基本安全需求在网格计算过程中所有安全需求中具有原子性的特点，即通过基本安全需求的逻辑定义我们可以实现对所有安全需求的形式化描述。其逻辑定义如下：

(1) 对应性

对应性安全需求基于事件的发生次序。在网格计算过程中，当一个事件 $Event_1$ 发生时，对应性需要确保另一事件 $Event_2$ 发生过。其可以被表示成如下的逻辑公式 F：

$Event_1 \Rightarrow Event_2$，其中，$Event_1$，$Event_2 \in EVENT$。

(2) 秘密性

秘密性一般需要建立在参与计算的各角色扮演者之间已经相互认证的基础之上，它需要确保网格计算参与者 Actor 除了直接的猜测以外没有获知消息数据 m 的能力。其可以被表示成如下的逻辑公式 F：

Comprise(m，Actor)，其中，$m \in M$，$Actor \in A$。

(3) 时效性

网格计算进行过程中，数据项 m 具有生存时间 Lifetime。其可以被表示成如下的逻辑公式 F：

Timeout(m，Lifetime)，其中，$m \in M$，$Lifetime \in T$。

(4) 公平性

在网格计算过程的任何阶段，计算参与主体中任何一方中止计算或者误操作，都不会使其它诚实主体的利益受到损失。

首先作如下定义：

PM ={<Actor，Mes>|$Actor \in A$,$Mes \subseteq 2^M$ }，其中元组<Actor，Mes> 表示根据公平性安全需求，在网格计算过程中计算参与主体与其利益的对应关系。

在网格计算过程中，如果 PM 集合中任意 Actor 成员的当前知识集包含代表其利益的 Mes 集合中的所有数据，公平性则需要确保 PM 集合中所有其它 Actor 成员的当前知识集也都包含其对应 Mes 集合中的所有数据。其可以被表示成如下的逻辑公式 F：

Fairness(PM)。

(5) 不可否认性

在一个事件 Event 发生后，通过角色扮演者组 Actors 中的成员能够合作证明这次事件的发生。其可以被表示成如下的逻辑公式 F：

Non_Repudiation(Event，Actors)，其中，Event$\in$*EVENT*，Actors$\in 2^{A}$。

(6) 匿名性

角色扮演者 $Actor_1$ 在参与网格计算过程中，其身份对于角色扮演者 $Actor_2$ 来说是匿名的。其可以被表示成如下的逻辑公式 F：

Anonym($Actor_1$，$Actor_2$)，其中，$Actor_1$，$Actor_2\in A$。

(7) 原子性

网格计算过程中对于网格计算任务组中的所有网格计算任务要么全部完成要么就全部失败。

首先作如下定义：

$ACT\subseteq CT$ 表示原子性要求中涉及的网格计算任务组。

原子性可以被表示成如下的逻辑公式 F：Atomicity(*ACT*)。

3.4 新模型的分析

本章提出了基于角色和任务的网格计算多用户协同关系描述模型 TRBCR，而且在其基础上构建网格计算安全需求分析模型 TRBCC。它与传统的安全需求分析模型相比具有以下特点：

(1) 在网格计算中最能体现网格的特点，同时安全需求最为复杂的是“多用户参与的协同计算”。TRBCR 通过引入网格计算任务及角色扮演者的概念，实现了对网格动态授权环境下的复杂协同关系的描述，从而为进一步通过 TRBCC 进行安全需求分析奠定了基础。

(2) TRBCC 通过形式化安全需求描述语言 TRBCL 实现了网格动态授权环境下多用户协同计算的不同安全需求的形式化描述。

4 一种基于虚拟组织的网格安全协议形式化新验证方法

本章论述一种基于虚拟组织的网格安全协议形式化验证方法。其基本思想是：通过在传统 Strand Space 理论的基础上引入网格计算信道的概念，把安全协议通信信道与虚拟组织中网格计算任务联系起来，并且近一步提出了一种基于虚拟组织的网格安全协议形式化验证方法，最终实现了网格环境下多用户协同计算安全协议的形式化分析与证明。

4.1 安全协议的形式化分析方法

安全协议提供安全服务，是保证系统安全性的基础。但是，设计一个符合系统安全目标的安全协议是十分困难的。因此我们必须借助形式化的方法，对安全协议进行设计和分析。自 20 世纪 70 年代末期 Dolev-Yao 模型[18]被提出以来，安全协议的研究已经成为一个热点，有众多的形式化研究方法涌现出来，其中主要有：

(1) 基于知识与信念推理的模态逻辑方法

模态逻辑方法是分析安全协议最直接最简单的一种方法。它们由一些命题和推理规则组成，命题表示主体的知识或信念，而应用推理规则可以从已知的知识和信念推导出新的知识和信念。

在这类方法中，比较有名的有：BAN 逻辑[19]，GNY逻辑[20]，AT逻辑[21]，VO逻辑[22]及SVO逻辑[23]，Bieber逻辑[24]，Syverson逻辑[25]，Rangan逻辑[25]， Moser逻辑[27]，Yahalom，Klein和Beth的YHK逻辑[28]以及Kessler和Wedel的AUTOLOG逻辑[29]等。

(2) 基于定理证明的分析方法

这种方法可以分为两类，一类是推理构造方法，另一类是证明构造方法。

推理构造方法主要包括： Meadows的NRL协议分析器方法[30]， Cervesato等学者的基于线性逻辑的协议验证方法[31]，Millen等学者的基于逻辑规则的协议验证方法[32]。

Kemmerer等学者研制的Ina Jo和ITP[33]是证明构造方法的典型代表。这一领域的另一项重要工作是Paulson的基于归纳的定理证明方法[34,35]。他研制的定理证明器Isabelle可以应用归纳方法分析安全协议。

(3) Spi演算方法

这种方法[36]根据Dolev-Yao模型，假定协议执行的每一步都可能与攻击者的执行步骤交叉。Pi演算是并发计算的基础，它引入了通道和作用域的概念。由于作用域之外的进程不能访问通道，在一定程度上保证了通道通信的安全性。Spi演算对pi演算进行了增强与扩充，增加了支持密码系统的原语，使Spi演算可以描述基于密码系统的安全协议。

4.2 扩展的Strand Space理论

Strand Space 理论是由 Thaye、Herzog 和 Guttman[37-39]在 1998 年提出的一种安全协议形式化分析方法，该方法吸纳了 NRL 协议器、Schneider 秩函数[40]和 Paulson 归纳法等思想，模型使用一种节点间存在因果关系的有向图来表示协议的运行，是分析安全协议的一种实用、直观和严格的形式化方法。 D.Song[41]对 Strand Space 模型进行了扩展，并开发了安全协议自动验证工具 ATHENA，实现了对认证性安全协议的自动验证。现在，Strand Space 理论已经渗透到了协议分析的方方面面，是近年出现的专用于协议分析的最有效的理论之一。

在网格计算环境下，虚拟组织是要建立在网格计算任务之上的，用户通过参与网格环境下的计算任务实现对虚拟组织中网格资源的访问。所以网格安全需求主要集中在多用户共同参与的网格计算过程。为此首先给出以下定义：

定义4.1. 网格计算任务是用来表示网格环境下的多用户协同计算的逻辑单元，它包括一系列可区分的动作，可能与多个用户相关，也可能包括几个子任务。对于安全协议P，其所涉及网格计算任务的集合记为CT^P。对于网格计算任务Ct_1，Ct_2，如果Ct_2是Ct_1的子任务，则可表示为$Ct_1 \angle Ct_2$。

定义 4.2. 协同计算组是由特定的网格计算任务的参与者所组成的集合。协同计算组的概念集中体现了基于虚拟组织的网格计算的显著特点，即动态性、多样性和自适应性，是网格虚拟组织形成的基础。

扩展的 Strand Space 理论主要由以下几部分组成：消息代数空间，为协议的规范提供基本项；Strand Space，主要提供协议规范及攻击行为的模型，是整个理论的主体部分之一；丛空间，经过对 Strand Space 的赋图结构，所有满足一定性质的子图的集合就构成了丛空间，这是主要的分析对象；构造攻击丛，即找出丛空间中与攻击行为有关的丛；构造虚拟组织内网格计算任务之间的攻击关系，即由于攻击者通过合法参与网格计算任务并且利用网格计算任务之间穿插运行实现对协议的攻击，从而导致网格计算任务之间产生的关联关系。图 4.1 说明了扩展的 Strand Space 理论的这些组成部分之间的关系。

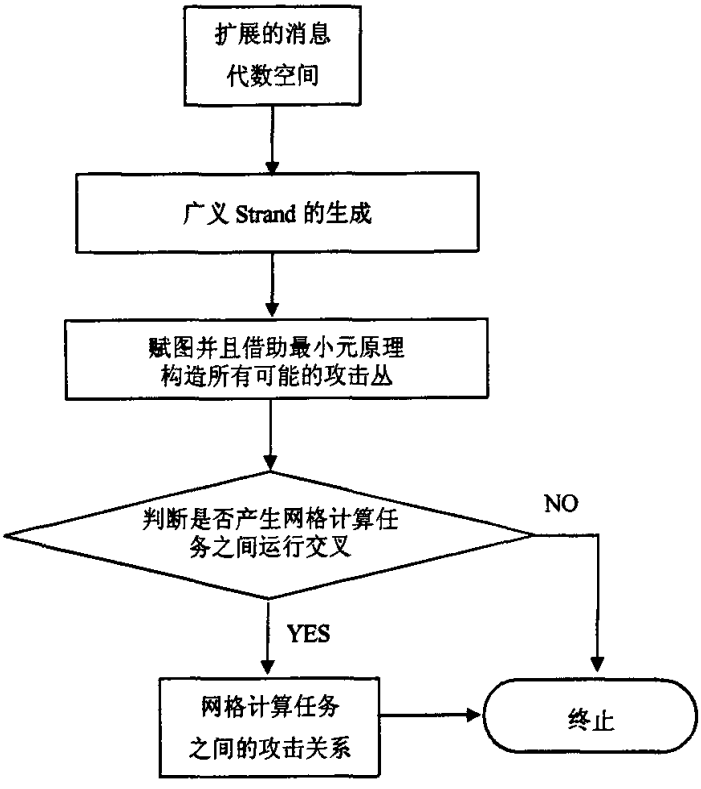


图 4.1 扩展的 Strand Space 理论结构示意图

Fig. 4.1 The structure of extended Strand Space theory

4.2.1 扩展的消息代数空间

在传统的网络环境下，协议描述中的消息包括：用户标识、随机数字、密钥等原子消息以及在原子消息上通过组合、加密、取消息摘要等运算得到的消息。在网格环境下，采用了证书机制，证书在网格环境下的安全协议中可以被作为原子消息处理。本文将基于网格环境下证书机制的特点构造扩展的消息代数空间。

定义 4.3. 原子消息代数空间 $MData_0$ 由以下几部分组成：

(1) 主体标识集合 T，主体包括协同计算组成员和可信第三方。

(2) 密钥集合 K，它分为公钥集合 PK 及私钥集合 SK。

(3) 新鲜随机数集合 N。

(4) 证书集合 *Cer*。

定义 4.4. 算子用来表示在原子消息上进行的运算。算子由以下几部分组成：

(1) Inv(k)：求逆密钥算子；

(2) Join(m_1,m_2)：联结算子；

(3) Encr(m,k)：加密算子；

(4) Sign(m,k)：签名算子；

(5) Hash(m,k)：单向散列函数算子。

今后记 Inv(k)为 k^{-1}，Encr(k,m)为$\{m\}_K$，Join(m,n)为 mn，与协议 P 相关的密码算子集合记为 $CryptFun^P$。

定义 4.5. 以原子消息代数空间 $MData_0$ 为基础，与协议 P 相关的消息代数空间 $MData^P$ 可被递归的定义如下：

(1) Inv(k)：如果 $\alpha \in MData_0$，那么 $\alpha \in MData^P$；

(2) 如果 $m_1 \in MData^P$，$m_2 \in MData^P$，那么 $m_1m_2 \in MData^P$；

(3) 如果 $m \in MData^P$，$k \in K$，$\sigma \in CryptFun^P$，那么 $\sigma(m,k) \in MData^P$。记密文的集合为 E，并称 $MData_0 \cup E$ 中的元素为素项。

定义 4.6. 子项关系$\Subset$递归地定义如下：

(1) Inv(k)：$a \Subset a$；

(2) $a \Subset \{g\}_k$ 若 $a \Subset g$；

(3) $a \Subset gh$ 若 $a \Subset g$ 或者 $a \Subset h$。

定义 4.7. 以协议 P 相关的消息空间 $MData^P$ 为基础，定义相关函数如下：

(1) Kpb(x)：$T \rightarrow PK$，返回主体 x 的公钥。

(2) Kpv(x)：$T \rightarrow SK$，返回主体 x 的私钥。

(3) Ses(x，y)：$T \times T \rightarrow K$，返回主体 x 与主体 y 之间的会话秘钥 k。

(4) Cert(CCSid，x)：$T \times T \rightarrow Cer$，主体 x 由可信第三方 CCSid 颁发的公钥证书。

4.2.2 广义角色的定义

协议由不同的协议角色组成，这已是共有的事实。但在网格计算过程中，主体之间是通过参与网格计算任务实现信息传递的，并且通过 SSL 协议，实现了协同计算组内部的信息共享以及协同计算组内部与外部的信息隔离。由此，本文将根据网格计算过程的特点提出新的广义角色的构造方法。

定义 4.8. 网格计算信道表示在网格计算过程中，主体之间通过参与网格计算任务 Ct 传递消息 α。记为 Channel(Ct，α)：$Ct \in CT$，$\alpha \in MData^P$。

定义 4.9. 协议的角色由一序列的通信步骤组成，每一个步骤有形式<ε,α>，这里 ε∈ {+,-}，α∈ $MData^P$，当 ε 是+时，<ε,α>表示发送消息 α；当 ε 是-时，<ε,α>表示接收消息 α。

定义 4.10. 设协议 P 由 n 个角色<$Role_1$, $Role_2$,..., $Role_n$>组成，又设角色 $Role_i$=

<<ε_{i1},α_{i1}>
<ε_{i2}, α_{i2}>
.
.
.
<ε_{ij}, α_{ij}>>

，那么该角色对应的广义角色 GenRole 可被定义为

<<ε_{i1}, Channel(Ct_{i1},α_{i1})>
<ε_{i2}, Channel(Ct_{i2},α_{i2})>
.
.
.
<ε_{ij}, Channel(Ct_{ij},α_{ij})>>

。

实际上，广义角色就是考虑传递消息所在协同计算组的角色。

4.2.3 攻击者知识集

设有限个网格虚拟组织成员 $P_1P_2\cdots P_n$ 参与执行安全协议 P，其中 P_e 为攻击者。在基于虚拟组织的网格体系架构下，每个参与网格计算任务的成员拥有对相关资源的使用权限，同时必须对其它成员提供相应资源。

(1) 初始知识集

在协议开始运行前，攻击者P_e的知识集称为初始知识集，设为INITIAL，INITIAL = *MData* ∪ *CT*，其中$MData \subseteq MData^P$，$CT \subseteq CT^P$。其中*MData*为攻击者所拥有的消息数据，CT为攻击者能够参与的网格计算任务。

(2) 动态的实体知识集

当协议开始执行以后，攻击者 P_e 所拥有的消息数据会发生变化。A_P 表示协议 P 运行过程中，攻击者 P_e 的当前知识集。

4.2.4 攻击者 Strand 的构造

攻击者-Strand 就是对攻击者能力的描述。Strand Space 模型用特定的 Strand 刻画了攻击者 8 个方面的能力：

(1) M. <+t> 其中t∈A；

(2) K. <+k> 其中k∈K_p(已泄露的密钥集合)；

(3) F. <-g>；

(4) T. <-g, +g, +g>；

(5) C. <-g, -h , +gh>；

(6) S. <-gh, +g, h>；

(7) E. <-k, -h, +$\{h\}_k$>；

(8) D. <-k^{-1}, -$\{h\}_k$, +h>；

在网格计算环境中，通过 SSL 协议建立协同计算组内部安全可靠的通信信道之后，就可以认为外部攻击者对通信信息的截取、窃听、伪造、修改等攻击是无效的；另一方面由于网格具有动态性特点，协议参与者内部信任关系是不稳定的。针对网格计算以上特点，构造如下攻击者-Strands：

(9) PD.〈-Channel(Ct,m)，-Ct，+m〉；

(10) PE.〈-m，-Ct，+Channel(Ct，m)〉。

以上攻击者-Strands 表明，在网格计算环境中，攻击者如要对通信信息进行攻击，必须首先经过授权加入相关通信信息所在的协同计算组。

4.2.5 与协议相关的 Strand Space 的构造

上一节定义了攻击者-Strand，现在我们来定义带攻击者的 Strand Space。

定义：把协议广义角色的角色名替换成实现该协议的主体名，这样生成的 Strand 称为正规-Strand。

定义：带攻击者的 Strand Space 由二部分组成：正规-Strand，攻击者-Strand。

所有的正规-Strand 都是由协议的诚实参与者生成；攻击者-Strand 为非诚实参与者中的系统合法用户，其攻击方式为：首先通过系统授权获得参与相关协同计算的能力，之后借助合法的协议运行获取攻击他人所需要的信息。

目前的 Strand Space 理论不能分析这样的攻击。本文将通过扩展 Strand Space 理论分析这种类型的攻击。

4.2.6 Strand Space 的附图结构

(1) 定义图的节点。

设 Strand s 为 $\begin{matrix}<<\varepsilon_{i1},Channel(Ct_{i1},\alpha_{i1})>\\ <\varepsilon_{i2},Channel(Ct_{i2},\alpha_{i2})>\\ \cdot\\ \cdot\\ \cdot\\ <\varepsilon_{ij},Channel(Ct_{ij},\alpha_{ij})>>\end{matrix}$，那么图的节点 n 被表示为<s,k>，它表示 s 的第 k 个分量。我们用 term(n)表示节点 n 上的消息项，即：$<\varepsilon_{ik},Channel(Ct_{ik},\alpha_{ik})>$。uns_term(n)表示节点 n 上没有符号的消息项，即 $Channel(Ct_{ik},\alpha_{ik})$。

(2) 定义图的边。

① 如果节点 n_1 和 n_2 满足 $term(n_1) = +Channel(Ct,\alpha)$，$term(n_2) = -Channel(Ct,\alpha)$，那么在这两个节点之间赋边→，即：$n_1 \rightarrow n_2$。

② 对于节点 n_1 和 n_2，如果存在广义 Strand 满足 $n_1 = <s,k>$，$n_2 = <s,k+1>$，那么在这两个节点之间赋边$\Rightarrow$，即：$n_1 \Rightarrow n_2$。

经过这样赋图结构后，Strand Space 变成了一个复杂的图，从这个复杂的图中截取满足一定性质的子图，这些子图包含了可能的运行情况，这些子图的全体就构成了我们的丛空间。下面我们就来分析这些子图应该满足的性质。

4.2.7 构造描述协议运行的数学对象：丛

丛是一个很好描述协议运行的概念，是对Dolev-Yao模型的带突破性的发展。下面的定义来自[37]。

定义4.11. 设$C=<N_C, (\rightarrow_C \cup \Rightarrow_C)>$是$<N, (\rightarrow \cup \Rightarrow)>$的子图。如果C满足下面的四个条件，则称之为丛(bundle)。

(1) C 有限。

(2) 如果 $n_2 \in N_c$ 且 term(n_2)是负的，则有唯一的一个 $n_1 \in N_c$，满足 $n_1 \rightarrow_C n_2$。

(3) 若 $n_2 \in N_c$ 且 $n_1 \Rightarrow n_2$，则 $n_1 \Rightarrow_C n_2$。

(4) C 是非循环的

若Strand s的所有节点都在C中，则称s在C中，可简记为$s \in C$。

4.2.8 借助极小元原理构造所有可能的攻击

构造攻击的基本方法是，构造出所有正规-Strand，之后取出每个正规-Strand；利用极小原理[37]构造出所有可能包含该 Strand 的丛，这些丛中包含了所有可能的攻击；根据攻击的定义找出所有的攻击丛，从而构造出相关攻击场景。

4.2.9 构造虚拟组织内网格计算任务之间的攻击关系

本节将考虑扩展的 Strand Space 理论中网格计算任务的代数特征，即：由于攻击者通过合法参与网格计算任务并且利用网格计算任务之间穿插运行实现对协议的攻击，从而导致网格计算任务之间产生的关联关系。首先对协议做如下定义：

定义 4.12. 对于协议 P，可以根据扩展的 Strand Space 理论的构造过程生成相应的网格计算任务集合 CT^P，消息代数空间 $MData^P$，Strand Space Σ^P 和丛空间 B^P，而且协议 P 唯一的确定了(CT^P, $MData^P$, $\sum^P$, B^P)，反过来，(CT^P, $MData^P$, Σ^P, B^P)将唯一确定协议 P，其中 CT^P 规定了协议所处的虚拟组织结构，$MData^P$ 规定了协议的消息结构，Σ^P 规定了协议的角色，而 B^P 反映了协议运行的情况，因此可以用(CT^P, $MData^P$, Σ^P, B^P)代表协议 P。

为了描述攻击场景中网格计算任务的代数特征，作如下定义：

定义 4.13. 设 C 是 B^P 中的丛，n_1，n_2 是 Σ_P 的节点并且 $n_1 \in N_c$，$n_2 \in N_c$。如果 n_1，n_2 满足下列条件之一：

(1) $n_1 \rightarrow n_2$；

(2) 存在广义 Strand s 满足 $n_1 = <s,k_1>$，$n_2 = <s,k_2>$，且 $k_1 < k_2$。

则记为 $n_1 \mapsto n_2$。

对于序列 $n_1 \mapsto n_2 \mapsto \cdots \mapsto n_k$，称为 C 中的一条路径。如果路径中含有攻击者-strand 节点，且除端点外所有节点均为攻击者-strand 节点，则称之为攻击路径。

由此，可以结合攻击场景对网格计算任务之间的攻击关系作如下定义：

定义 4.14. 设 Ct_1，Ct_2 是 CT^P 中不同的网格计算任务，C 是 B^P 中的攻击丛。如果 C 中存在节点 n_1，n_2 且满足下列条件：

(1) $term(n_1)$=+Channel(Ct_1，α_1)，$term(n_1)$=-Channel(Ct_2，α_2)；

(2) 存在 $m \in MData^P$ 满足 $m \in \alpha_1$，$m \in \alpha_2$；

(3) C 中存在以 n_1，n_2 为端点的路径。

则称攻击丛 C 产生关于网格计算任务 Ct_1，Ct_2 的攻击，记为 $Ct_1 \triangleright {}^C Ct_2$；如果对于攻击丛 C，同时存在 $Ct_1 \triangleright {}^C Ct_2$ 与 $Ct_2 \triangleright {}^C Ct_1$，则称攻击丛 C 产生关于网格计算任务 Ct_1，Ct_2 的交织型攻击，记为 $Ct_1 \rightleftharpoons {}^C Ct_2$。

最终，可以针对网格计算环境下安全协议作如下定义：

定义 4.15. (CT^P，$MData^P$，Σ^P，B^P，$\triangleright$，$\rightleftharpoons$) 可以有效表达网格计算环境下关于安全协议 P 的攻击行为。

4.3 新验证方法的分析

本章通过对传统 Strand Space 理论进行扩展，提出了一种基于虚拟组织的网格安全协议形式化验证方法，实现了网格环境下多用户协同计算安全协议的分析与证明。与传统的 Strand Space 理论相比，扩展之后内容如下：

(1) 对原有理论的消息空间进行了扩展，证书在安全协议中被作为原子消息处理。

(2) 通过引入网格计算信道的概念，把安全协议通信信道与虚拟组织中网格计算任务联系起来，并且进一步建立网格计算任务相对于安全协议的代数特征。

(3) 对原有理论中敌手模型进行了改进，使之能够描述敌手通过参与网格计算任务而对安全协议产生的攻击。

5 实例分析

本章将通过实例展示TRBCC网格安全需求分析模型以及扩展的Strand Space理论与传统安全模型的不同.

5.1 基于虚拟组织的网格计算场景

在网格协同计算环境下，通过采用基于虚拟组织的分布式管理模式，使得作业实体从资源控制、任务调度和管理的复杂工作中解脱出来。为了获得充分而必需的资源，各个VO可以使用标准的、开放的、通用的接口进行信息交互，并根据这些信息来协调各自的资源使用策略，从而避免系统的盲目查找和不合理的远程调用现象，以此大大提高了网格计算的智能性。在地域上分布的异构网格计算环境下，计算任务能自主地从某一计算节点迁移到另一计算节点，并可与其它VO的资源进行交互以实现作业和资源管理的自适应。

图5.1是基于虚拟组织的网格计算场景示意图。

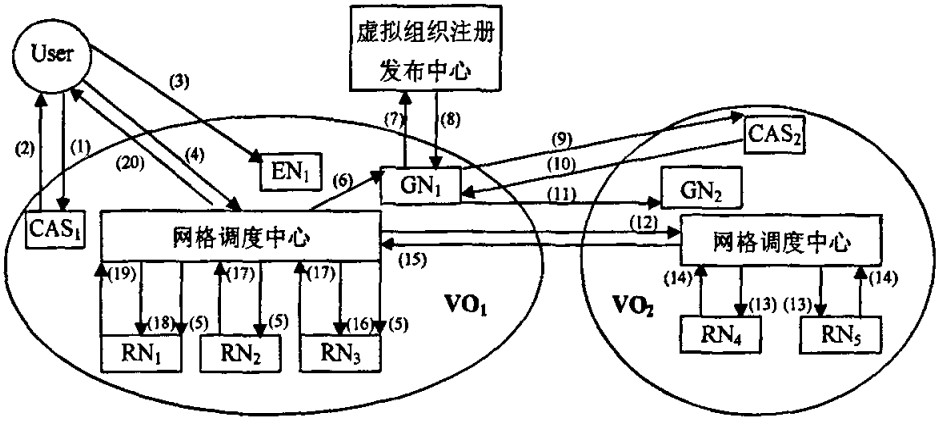


图5.1 网格协同计算场景示意图

Fig. 5.1 Grid coordinative computing process

在网格计算过程P中，用户User属于虚拟组织VO_1，RN_1，RN_2，RN_3是属于VO_1的资源，RN_4，RN_5是属于虚拟组织VO_2资源，可对图5.1的基于虚拟组织的网格协同计算过程作如下说明：

(1) 用户向 CAS_1 提供认证信息。

(2) 用户认证通过，CAS_1 对其进行授权，并进行电子签名，把信息返回给用户。

(3) 用户向 VO_1 的入口节点 EN_1 传递授权信息。

(4) 用户向虚拟组织 VO_1 调度中心提出作业要求。

(5) VO_1 调度中心获取本地网格资源 RN_1，RN_2，RN_3，并分别向其分配计算任务。

(6) VO_1 调度中心发现虚拟组织 VO_1 的资源无法完成任务，则向网关节点 GN_1 汇报。

(7) GN_1 向虚拟组织注册发布中心提出服务查询要求。

(8) 虚拟组织注册中心告诉 GN_1 有关虚拟组织 VO_2 的信息。

(9) GN_1 向 CAS_2 提供认证信息。

(10) GN_1 认证通过，CAS_2 其进行相应授权，并进行电子签名，把信息返回给 GN_1。

(11) GN_1 向 VO_2 的网关节点 GN_2 传递授权信息。

(12) VO_1 调度中心向 VO_2 的调度中心提出作业要求。

(13) VO_2 调度中心选取合适资源 RN_4，RN_5，并向 RN_4，RN_5 分配任务。

(14) RN_4，RN_5，完成任务，向 VO_2 任务调度中心返回运行结果。

(15) VO_2 任务调度中心向 VO_1 任务调度中心返回运行结果。

(16) VO_1 任务调度中心将运行结果返回给 RN_3 资源代理。

(17) 资源 RN_2，RN_3 完成任务，向 VO_1 任务调度中心返回运行结果。

(18) 任务调度中心将 RN_2，RN_3 任务运行结果传递给 RN_1。

(19) 资源 RN_1 利用从 VO_1 任务调度中心得到的数据完成任务，并向 VO_1 任务调度中心返回运行结果。

(20) VO_1 任务调度中心将运行结果交给用户 User。

5.2 网格计算对应 TRBCC 空间

网格计算对应 TRBCC 安全空间如下：

TRBCR=（U，R，T，A，CT，RM，CA，M，Cer）其中，

U = {User，RN_1，RN_2，RN_3}；

R = {R_User，R_RN_1，R_RN_2，R_RN_3}；

A = {Actor_User，Actor_RN_1，Actor_RN_2，Actor_RN_3}；

CT = {Ct_1，Ct_2}；

RM = {Rm_ CAS_1}；

CA = {Ca}；

其对应 AT ，RMA 关系如图 5.2，图 5.3 所示。

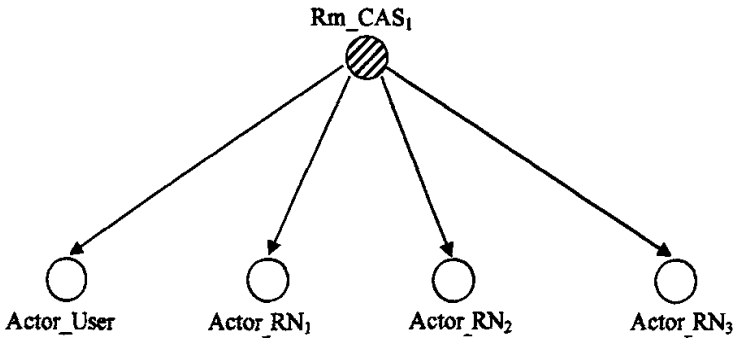


图 5.2 TRBCC 中 *RMA* 关系

Fig. 5.2 *RMA* relation in TRBCC

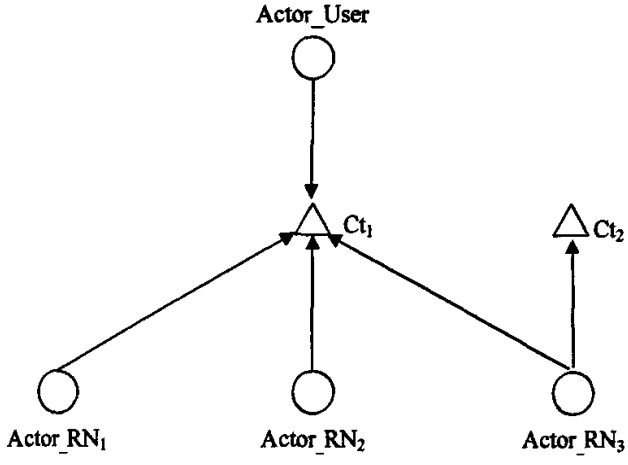
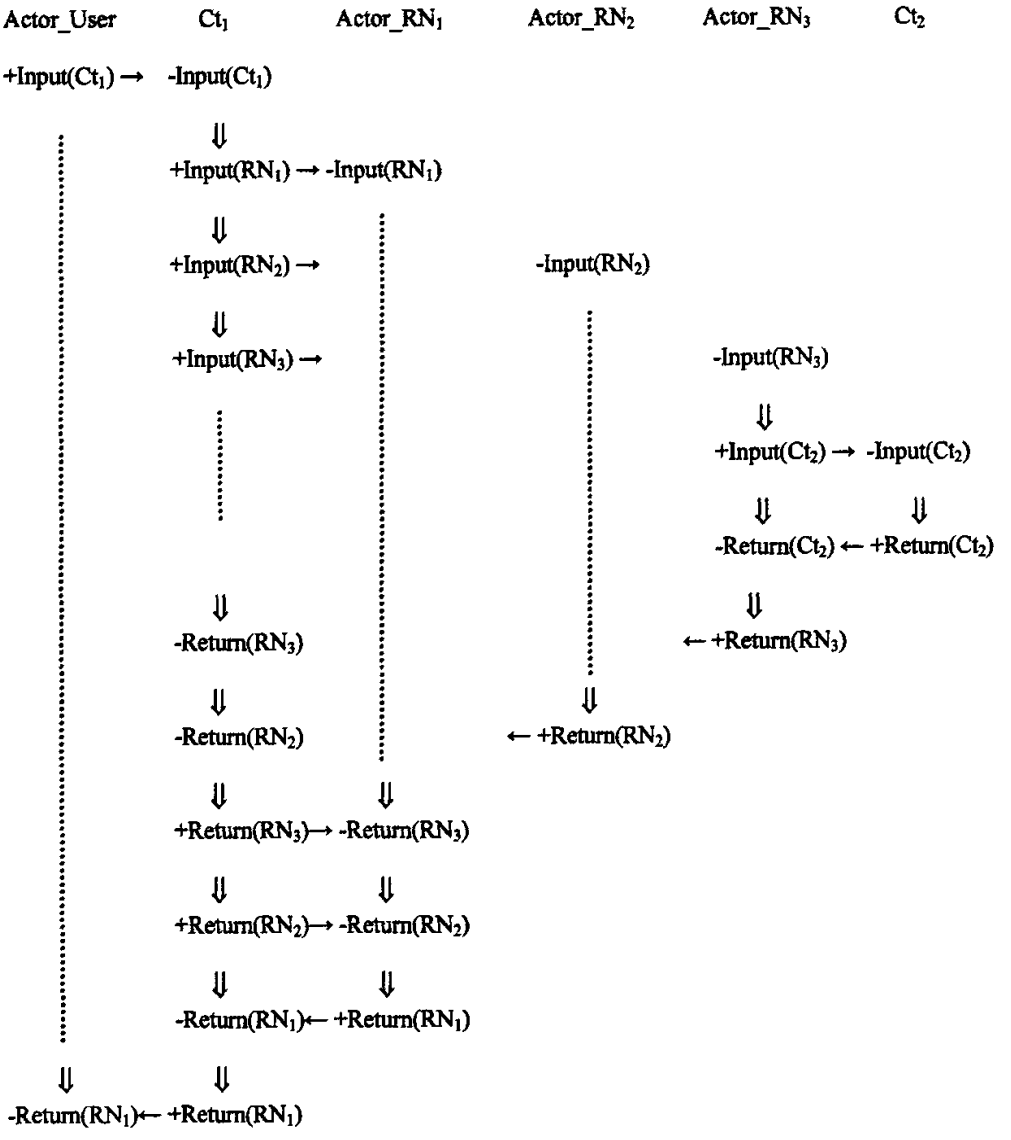


图 5.3 TRBCC 中 *AT* 关系

Fig. 5.3 *AT* relation in TRBCC

另外，TRBCC 安全空间对应的网格计算信息交互图如下，其中：消息数据集合 M = {Input(Ct_1)，Input(RN_1)，Input(RN_2)，Input(RN_3)，Input(Ct_2)，Return(Ct_2)，Return(RN_3)，Return(RN_2)，Return(RN_1)}，Input(Ct_1)表示用户提交的网格计算任务描述，Input(Ct_2)表示 VO_1 向 VO_2 提交的网格计算任务描述，Input(RN_1)、Input(RN_2)、Input(RN_3)分别表示资源 RN_1、RN_2、RN_3 从 VO_1 任务调度中心得到的输入数据，Return(Ct_2)表示 VO_2 返回的计算结果，Return(RN_3),Return(RN_2)，Return(RN_1)分别表示资源 RN_1、RN_2、RN_3 返回的计算结果。



同时，针对网格计算任务 Ct_2可以得到 TRBCC 安全空间的子空间 TRBCC'，并且满足 TRBCC'$\angle_{ct2}$ TRBCC。其中：

TRBCR'=（*U*，*R*，*T*，*A*，*CT*，*RM*，*CA*，*M*，*Cer*）；

U = {RN_3，RN_4，RN_5}；

R = {R_RN_3，R_RN_4，R_RN_5}；

A = {$Actor\_RN_3$，$Actor\_RN_4$，$Actor\_RN_5$}；

CT = {Ct_{21}，Ct_{22}}；

RM' = {Rm_CAS_2}；

CA={Ca'}；

其对应 *AT'*，*RMA* 关系如图 5.4，图 5.5 所示。

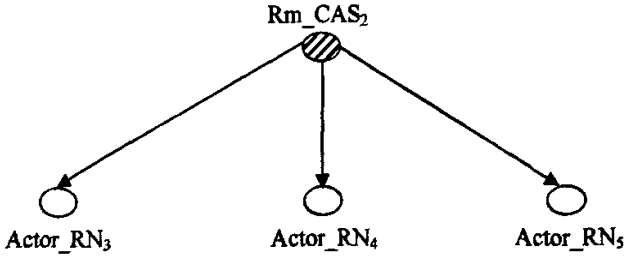


图 5.4 TRBCC'中 *RMA* 关系

Fig. 5.4 *RMA* relation in TRBCC'

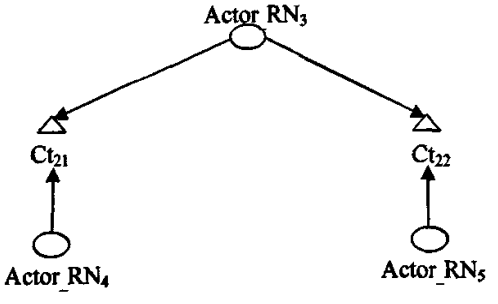


图 5.5 TRBCC'中 *AT*关系

Fig. 5.5 *AT* relation in TRBCC'

5.3 网格计算安全需求形式化描述

在 TRBCC 安全空间中对网格计算个别安全需求进行形式化描述如下：

(1) 具有角色 R_User 的用户加入网格计算任务 Ct1 之前需要与参与网格计算任务 Ct_1 的资源之间进行双向认证。

RANGE TRBCC

GET A($Actor_1$，$Actor_2$): $Actor_1$.Role=R_User ∧ ¬ ($Actor_2$.Role=R_User)

SECURITY REQUIREMENTS: Author（Rm_CAS_1，$Actor_1$，Ct_1，Lifetime）⇒ Authen($Actor_1$，$Actor_2$，CERT(Ca，$Actor_2$.User)) ∧ Authen($Actor_2$，$Actor_1$，CERT(Ca，$Actor_1$.User))

(2) 资源节点 RN_2 与 RN_3 之间对于从调度中心处获得的输入信息数据具有公平性。

RANGE TRBCC

GET A($Actor_1$，$Actor_2$) $Actor_1$.Role=R_ RN_2 ∧ $Actor_2$.Role=R_ RN_3

SECURITY REQUIREMENTS: Fairness({<$Actor_1$，{Input(RN_2)}>，<$Actor_2$，{Input(RN_3)}>})

(3) 用户 User 从资源节点 RN_1 获得信息数据 Return(RN_1)对于资源节点 RN_1 具有不可否认性。

RANGE TRBCC

GET A($Actor_1$，$Actor_2$): $Actor_1$.Role=R_User ∧ $Actor_2$.Role=R_ RN_1

SECURITY REQUIREMENTS: Non_Repudiation(Receive($Actor_1$,$Actor_2$,Return(RN_1)), {$Actor_2$})

(4) 网格计算任务 Ct_{21}，Ct_{21} 具有原子性。

RANGE TRBCC'

SECURITY REQUIREMENTS: Atomicity ({Ct_{21}，Ct_{21}})

5.4 安全协议描述

网格计算在 TRBCC'安全空间具备以下安全需求：

在网格计算任务 Ct_{21} 执行过程中，资源 RN_3 与 RN_4 之间需要进行双向认证。同时，在网格计算任务 Ct_{22} 执行过程中，资源 RN_3 与 RN_5 之间也需要进行双向认证。

由此，双向认证对应安全协议 P 如下：

(1) A →B：Cert(Ca, A)

(2) B →A：Cert(Ca, B)

(3) A →B：$\{M, A\}_{Kpb(B)}$

(4) B →A：$\{M, N\}_{Kpb(A)}$

(5) A→B：$\{N\}_{Kpb(B)}$

下一步则要构建安全协议 P 所对应的 Strand Space。

5.5 Strand Space构建

安全协议 P 中有两个角色：INIT 和 RESP。其对应广义角色分别为：

INIT[Ct,A,B,M,N]	RESP[Ct,A,B,M,N]
1：<+Channel(Ct, Cert(Ca, A))>	1：<-Channel(Ct, Cert(Ca, A))
⇓	⇓
2：<-Channel(Ct, Cert(Ca, B))>	2：<+Channel(Ct, Cert(Ca, B))>
⇓	⇓
3：<+Channel(Ct, $\{M, A\}_{Kpb(B)}$)>	3：<-Channel(Ct, $\{M, A\}_{Kpb(B)}$)>
⇓	⇓
4：<-Channel(Ct, $\{M, N\}_{Kpb(A)}$)>	4：<+Channel(Ct, $\{M, N\}_{Kpb(A)}$)>
⇓	⇓
5：<+Channel(Ct, $\{N\}_{Kpb(B)}$)>	5：<-Channel(Ct, $\{N\}_{Kpb(B)}$)>

安全协议 P 对应 Strand Space 包括如下正规-Strand：

$INIT[Ct_{21}, RN_4, RN_3, N_{RN4}, N_{RN5}]$，$RESP[Ct_{22}, RN_4, RN_5, N_{RN4}, N_{RN5}]$ 。

其中，Ct_{21}、Ct_{22}分别表示虚拟组织 *VO* 中的网格计算任务，RN_3、RN_4、RN_5分别表示参与安全协议 P 的虚拟组织资源主体标识，N_{RN4}、N_{RN5}则分别对应资源主体 RN_4、RN_5在安全协议 P 执行过程中所生成的随机数。

5.6 实例化攻击场景构建

如果攻击者初始知识集 INITIAL = *MData* U *CT*，*MData* ={Kpb(RN_1)，Kpv(RN_1)}，*CT* ={Ct_{21}，Ct_{22}},则可以通过对 Strand Space 赋图并利用极小元原理构造攻击丛 C,从而构建出安全协议 P 的实例化攻击场景，如图 5.6 所示。

5.7 网格计算任务攻击关系构建

对于网格计算任务 C_{21}，C_{22}，攻击丛 C 中存在节点 n_1=<INIT[Ct_{21}, RN_4, RN_3, N_{RN4}, N_{RN5}]，1>，n_2=< RESP[Ct_{22}, RN_4, RN_5, N_{RN4}, N_{RN5}]，1>，其满足下列条件：

(1) term(n_1) =+Channel(Ct_{21} ， $\{N_{RN4},RN_4\}_{Kpb(RN3)}$)) ， term($n_2$)=-Channel($Ct_{22}$ ，$\{N_{RN4},RN_4\}_{Kpb(RN5)}$))；

(2) 存在消息数据$\{N_{RN4}, RN_4\} \in MData^P$，并且满足：$\{N_{RN4}, RN_4\} \Subset \{N_{RN4}, RN_4\}_{Kpb(RN3)}$)，$\{N_{RN4}, RN_4\} \Subset \{N_{RN4}, RN_4\}_{Kpb(RN5)}$)；

(3) C 中存在以 n_1，n_2 为端点的路径。

由此可以得出结论：$C_{21} \triangleright^{C} C_{22}$；同理可得出结论：$C_{22} \triangleright^{C} C_{21}$，所以攻击丛 C 产生关于网格计算任务 Ct_{21}，Ct_{22} 的交织型攻击，记为 $C_{21} \rightleftharpoons^{C} C_{22}$。

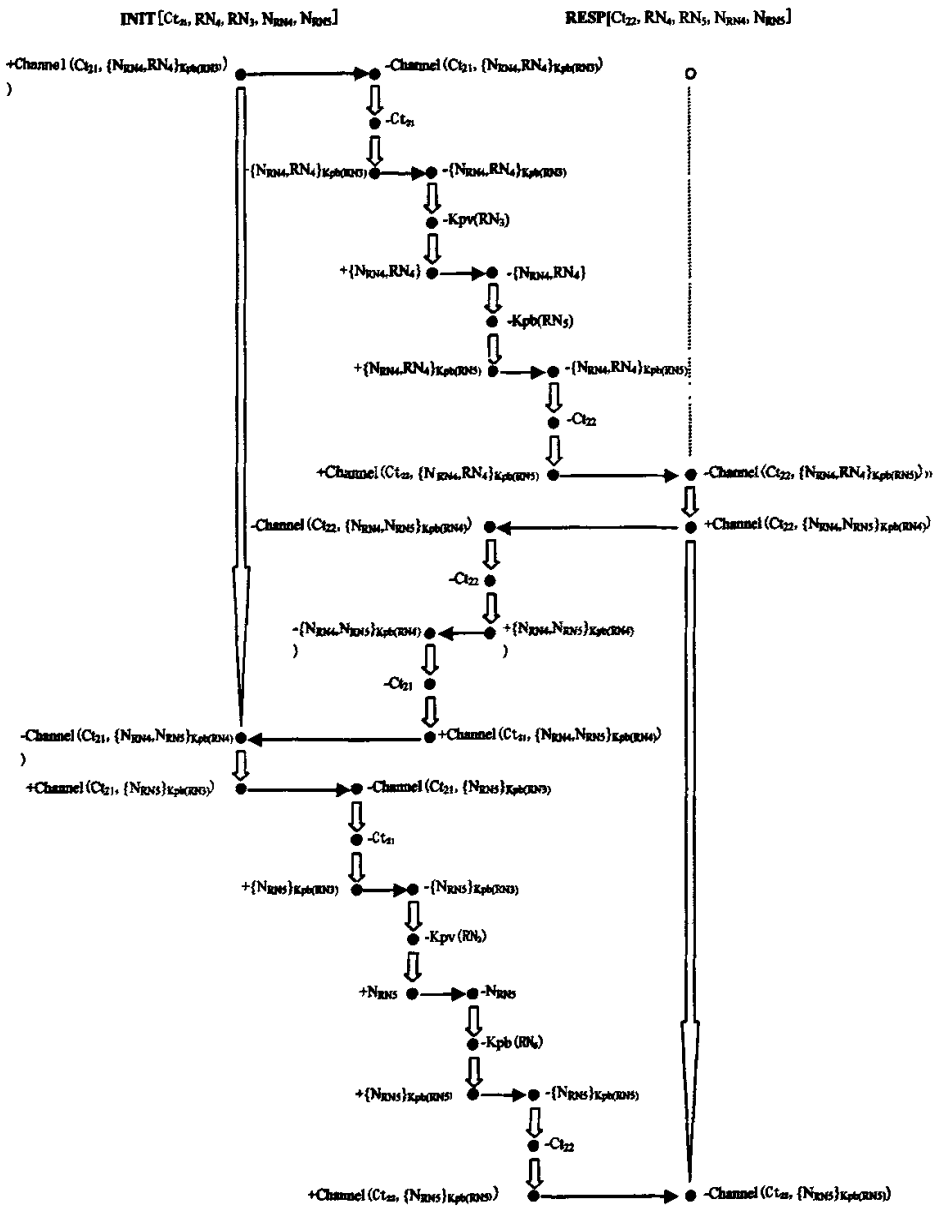


图 5.6　实例化攻击场景示意图

Fig. 5.6 Instantiation of attack process

结　论

本文在分析了网格环境下多用户参与的协同计算安全需求的基础上提出了一个基于角色和任务的网格计算安全需求分析模型 TRBCC，实现了网格环境下多用户参与的协同计算安全需求的形式化描述。

TRBCC 安全空间是基于角色与任务的网格安全需求分析模型。该模型由三部分组成：网格计算多用户协同关系描述模型 TRBCR、网格计算信息交互图和网格安全需求的形式化描述语言。TRBCC 模型建立过程如下：

根据网格计算任务描述首先建立起网格计算多用户协同关系描述模型 TRBCR，对在网格动态授权环境下的协同关系进行描述；然后在 TRBCR 模型基础上构建网格计算信息交互图，形成对网格计算任务参与者之间消息数据交互和处理过程的描述；并且进一步根据本文提出的一种网格安全需求的形式化描述语言，完成对网格动态授权环境下安全需求的形式化描述。

其中，TRBCR 协同关系描述模型引入了角色扮演者和网格计算任务以及角色管理者的概念，使之能够准确的描述网格动态授权环境下的协同关系，通过在其基础上建立的网格安全需求的形式化描述语言 TRBCL，可以精确的把握网格系统的安全需求，从而可以有效的布置安全策略，设计安全协议。

同时，本文通在传统 Strand Space 理论的基础上提出了一种基于虚拟组织的网格安全协议形式化验证方法，实现了网格环境下多用户协同计算安全协议的形式化分析与证明。

扩展的 Strand Space 理论主要由以下几部分组成：消息代数空间，为协议的规范提供基本项；Strand Space，主要提供协议规范及攻击行为的模型，是整个理论的主体部分之一；从空间，经过对 Strand Space 的赋图结构，所有满足一定性质的子图的集合就构成了从空间，这是主要的分析对象；构造攻击丛，即找出从空间中与攻击行为有关的丛；构造虚拟组织内网格计算任务之间的攻击关系，即由于攻击者通过合法参与网格计算任务并且利用网格计算任务之间穿插运行实现对协议的攻击，从而导致网格计算任务之间产生的关联关系。

与传统的 Strand Space 理论相比，扩展之后内容对原有理论的消息空间进行了扩展，证书在安全协议中被作为原子消息处理；通过引入网格计算信道的概念，把安全协议通信信道与虚拟组织中网格计算任务联系起来，并且进一步建立网格计算任务相对于安全协议的代数特征；对原有理论中敌手模型进行了改进，使之能够描述敌手通过参与网格计算任务而对安全协议产生的攻击。

在未来的工作中，我们将把 TRBCC 安全需求分析模型以及扩展的 Strand Space 理论引入到网格安全移动代理的设计中来，最终实现一个在动态网格计算环境下的保证协作安全和服务可靠性的安全中间件。

参 考 文 献

[1] Whitfield Diffie, Paul C, van Oorschot, Michael J. Wiener. Authentication and Authenticated Key Exchanges. Designs, Codes and Cryptography. 1992, 17(2):107-125.

[2] M. Bellare, P. Rogaway. Entity Authentication and key distribution. In:Advances in Cryptology-Crypto 93 Proceedings. 1994:173-188.

[3] Paul F. Syverson and Paul C. van Oorschot. On Unifying Some Cryptographic Protocol Logics. In:Proceedings of the IEEE Computer Society Symposium on Research in Security and Privacy. 1994:14-28.

[4] Woo T., Lam S..A semantic model for authentication protocols. In:Proceedings of the IEEE CS Symposiumon Research in Security and Privacy. 1993:178-194.

[5] Syverson P, Meadows C. A logical language for specifying cryptographic protocol requirements. In:Proceedings of the IEEE CS Symposium on Research in Security and Privacy. 1993:165-177.

[6] Frank Siebenlist. Grid security:requirements, plans and ongoing efforts. Presented at ACM workshop on XML security, Fairfax, Virginia, 2003.

[7] Foster I, Kesselman C. The Anatomy of the Grid:Enabling Scalable Virtual Organizations. International Joural Supercomputer Applications. 2001, 15(3):200-222.

[8] Foster I, Kesselman C. The Grid:Blueprint for a New Computing Infrastructure. Morgan Kaufmann, 1998.

[9] Foster I, Berry D, Djaout A. The Open Grid Services Architecture Version 1.0. http://forge.gridforum.org/projects/ogsa -WG, 2004.

[10] Tuecke S, Czajkowski K, Foster I. Grid Service Specification. Http://www.Globus.Org/research/papers, 2002.

[11] WS-Resource Framework. Http://www.oasis-open.org/apps/org/workgroup/wsrf/.

[12] Http://www.globus.org.

[13] Butler R, Engert D, Foster I. A National-Scale Authentication Infrastructure[J]. IEEE Computer. 2000:33(12):60-66.

[14] Foster I, Kesselman C, Tsudik G. A Security Architecture for Computational Grids[C]. ACM conference on Computers and Security, 1998:83-91.

[15] Globus Security Policy and Implementation[EB/OL]. Http://www.globus.org/security, 1997.

[16] Pearlman L, Welch V, Foster I. A Community Authorization Service:Status and Future[C]. CHEP, 2003.

[17] Pearlman L, Welch V, Foster I. A Community Authorization Service for Group Collaboration[C]. IEEE the 3rd International Workshop on Policies for Distributed Systems and Network, 2002.

[18] Dolev D, Yao A. On the security of public key protocols. IEEE Transactions on Information Theory. 1983, 29(2):198-208.

[19] Burrows M, Abadi M, Needham R, A logic of authentication. In:Proceedings of the Royal Society of London A. 1989, 426(2):233-271.

[20] Gong L, Needham R, Yahalom R. Reasoning about belief in cryptographic protocols. In: Proceedings of the 1990 IEEE Computer Society Symposium on Research in Security and Privacy. Los Alamitos:IEEE Computer Society Press. 1990:234-248.

[21] Abadi M. Tuttle MR. A semantics for a logic of authentication. In:Proceedings of the 10th ACM Symposium on Principles of Distributed Computing. ACM Press. 1991:201-216.

[22] van Oorschot PC. Extending cryptographic logics of belief to key agreement protocols. In: Proceedings of the 1st ACM Conference on Computer and Communications Security. ACM Press. 1993:233-243.

[23] Syverson PF, van Oorschot PC. On unifying some cryptographic protocol logics. In: Proceedings of the 1994 IEEE Computer Society Symposium on Research in Security and Privacy. 1994:14-28.

[24] Bieber P. A Logic of Communication in a Hostile Environment. In:Proceedings of the Computer Security Foundations Workshop III. 1990:14-22.

[25] Syverson P. Formal semantics for logics of cryptographic protocols. In:Proceedings of the Computer Security Foundations Workshop III. 1990:32-41.

[26] Rangan PV. An axiomatic basis of trust in distributed systems. In:Proceedings of the 1988 Symposium on Security and Privacy. 1988:204-211.

[27] Moser L. A logic of knowledge and belief for reasoning about computer security. In: Proceedings of the Computer Security Foundations Workshop II. 1989:57-63.

[28] Yahalom R, Klein B, Beth T. Trust relationships in secure systems:A distributed authentication perspective. In:Proceedings of the 1993 IEEE Symposium on Security and Privacy. 1993:150-164.

[29] Kessler V, Wedel G. AUTOLOG-An advanced logic of authentication. In:Proceedings of the Computer Security Foundations Workshop. 1994:90-99.

[30] Meadows C. The NRL protocol analyzer:An overview. Journal of Logic Programming. 1996, 26(2):113-131.

[31] Cervesato I, Durgin N, Lincoln P, Mitchell J. A meta-notation for protocol analysis. In: Proceedings of the 1999 IEEE Symposium on Security and Privacy. Los Alamitos:IEEE Computer Society Press. 1999:55-69.

[32] Millen J. The Interrogator model. In:Proceedings of the 1995 IEEE Symposium on Security and Privacy. Los Alamitos:IEEE Computer Society Press, 1995:251-260.
[33] Kemmerer R, Meadows C, Millen J. Three systems for cryptographic protocol analysis. Journal of Cryptology. 1994, 7(2):251-260.
[34] Paulson LC. Mechanized proofs for a recursive authentication protocol. In:Proceedings of the 10th IEEE Computer Security Foundations Workshop. Los Alamitos:IEEE Computer Society Press, 1997:84-94.
[35] Paulson LC. The inductive approach to verifying cryptographic protocols. Journal of Computer Security:1998, 9(6):85-128.
[36] Abadi M, Gordon AD. A calculus for cryptographic protocols:The spi calculus. In: Proceedings of the 4th ACM Conference on Computer and Communications Security. 1997:36-47.
[37] Thayer FJ, Herzog JC, Guttman JD. Strand spaces:Why is a security protocol correct? In:Proceedings of the 1998 IEEE Symposium on Security and Privacy. Los Alamitos:IEEE Computer Society Press, 1998:160-171.
[38] Thayer FJ, Herzog JC, Guttman JD. Strand spaces:Proving security protocols correct. Journal of Computer Security. 1999, 7(2-3):191-230.
[39] Thayer FJ, Herzog JC, Guttman JD. Strand spaces:Honest ideals on strand spaces. In: Proceedings of the 1998 IEEE Computer Security Foundations Workshop. Los Alamitos:IEEE Computer Society Press, 1998:66-77.
[40] Schneider S. Verifying authentication protocols with CSP[A]. In:Proceedings of the 10th IEEE Computer Security Foundations Workshop[C]. IEEE Computer Society Press, 1997:3-17.
[41] Song D. Athena:A new efficient automatic checker for security protocol analysis. In: Proceedings of the 1999 IEEE Computer Security Foundations Workshop. Los Alamitos:IEEE Computer Society Press, 1999:192-202.
[42] 郁志辉，陈渝，刘鹏. 网格计算[M]. 北京:清华大学出版社, 2002.
[43] Xu Feng, Xie Li etc.. Service-Oriented Role-Based Access Control. Chinese Journal of Computers. 2004, 28(4):687-693.
[44] Whitfield Diffie, Paul C, van Oorschot, Michael J. Wiener. Authentication and Authenticated Key Exchanges. Designs, Codes and Cryptography. 1992, 18(2):107-125.
[45] Syverson P, Meadows C. A logical language for specifying cryptographic protocol requirements. In:Proceedings of the IEEE CS Symposium on Research in Security and Privacy. 1993:165-177.
[46] QING Si-Han. Twenty Years Development of Security Protocols Research. Journal of Software. 2003, 14(10):1740-1752.
[47] Sandhu R, Conyne E. J, Lfeinstein H. L et al.. Role based access control models. IEEE Computer. 1996, 29(2):38-47.

[48] Ferraiolo D. F, Sandhu R, Guirila S, Kuhn D. R, Chandramouli R. Proposed NIST standard for role-based access control. ACM Transactions on Information and System Security. 2001, 4(3):224-274.

[49] A. Zieger. Grid security: state of the art. IBM developerworks, 2003.

[50] T. Myer. Grid watch:GGF and grid security vol. IBM developerworks, 2004.

[51] Frank Siebenlist, Foster I. Security for Grid Services. In:Proceedings of the 12th IEEE International Symposium on High Performance Distributed Computing. 2003:48.

[52] SSL v3.0 specification. http://home. netscape. com/eng/ssl3/3-SPEC. htm.

[53] Dolev D, Yao A. On the security of public key protocols. IEEE Transactions on Information Theory. 1983, 29(2):198-208.

[54] W. Marrero, Ed Clarke and Somesh Jha. Verifying security protocols with Brutus. ACM Transactions on software engineering and methodology. 2000, 9(4).

攻读硕士学位期间发表学术论文情况

一种基于虚拟组织的网格安全协议形式化验证方法，论文第四章，赵辉、李明楚，计算机工程与应用，已录用。

致　谢

在本文撰写过程中，本人导师给予了全面指导并且详细审阅了论文全稿。另外，杨斌、姜增虎等同学提出了许多有益的意见，在此向他们致以衷心的感谢。

基于角色和任务的网格安全需求分析模型

作者： 赵辉
学位授予单位： 大连理工大学

参考文献(54条)

1. Whitfield Diffie. Paul C. van Oorschot. Michael J.Wiener Authentication and Authenticat-ed Key Exchanges 1992(02)

2. M Bellare. P Rogaway Entity Authentication and key distribution 1994

3. Paul F Syverson. Paul C. van Oorschot On Unifying Some Cryptographic Protocol Logics 1994

4. Woo T. Lam S A semantic model for authentication protocols 1993

5. Syverson P. Meadows C A logical language for specifying cryptographic protocol requirements 1993

6. Frank Siebenlist Grid security:requirements, plans and ongoing efforts 2003

7. Foster I. Kesselman C The Anatomy of the Grid:Enabling Scalable Virtual Organizations 2001(03)

8. Foster I. Kesselman C The Grid:Blueprint for a New Computing Infrastructure 1998

9. Foster I. Berry D. Djaout A The Open Grid Services Architecture Version 1.0 2004

10. Tuecke S. Czajkowski K. Foster I Grid Service Specification 2002

11. WS-Resource Framework

12. 查看详情

13. Butler R. Engert D. Foster I A National-Scale Authentication Infrastructure 2000(12)

14. Foster I. Kesselman C. Tsudik G A Security Architecture for Computational Grids 1998

15. Globus Security Policy and Implementation 1997

16. Pearlman L. Welch V. Foster I A Community Authorization Service:Status and Future 2003

17. Pearlman L. Welch V. Foster I A Community Authorization Service for Group Collaboratio-n 2002

18. Dolev D. Yao A On the security of public key protocols 1983(02)

19. Burrows M. Abadi M. Needham R A logic of authentication 1989(02)

20. Gong L. Needham R. Yahalom R Reasoning about belief in cryptographic protocols 1990

21. Abadi M. Tuttle MR A semantics for a logic of authentication 1991

22. van Oorschot PC Extending cryptographic logics of belief to key agreement protocols 1993

23. Syverson PF. van Oorschot PC On unifying some cryptographic protocol logics 1994

24. Bieber P A Logic of Communication in a Hostile Environment 1990

25. Syverson P Formal semantics for logics of cryptographic protocols 1990

26. Rangan PV An axiomatic basis of trust in distributed systems 1988

27. Moser L A logic of knowledge and belief for reasoning about computer security 1989

28. Yahalom R. Klein B. Beth T Trust relationships in secure systems:A distributed authentication perspective 1993

29. Kessler V. Wedel G AUTOLOG-An advanced logic of authentication 1994

30. Meadows C The NRL protocol analyzer:An overview 1996(02)

31. Cervesato I. Durgin N. Lincoln P. Mitchell J A meta-notation for protocol analysis 1999

32. Mi llen J The Interrogator model 1995

33. Kemmerer R. Meadows C. Millen J Three systems for cryptographic protocol analysis 1994(02)

34. Paulson LC Mechanized proofs for a recursive authentication protoc01 1997

35. Paulson LC The inductive approach to verifying cryptographic protocols 1998(06)

36. Abadi M. Gordon AD A calculus for cryptographic protocols:The spi calculus 1997

37. Thayer FJ. Herzog JC. Guttman JD Strand spaces:Why is a security protocol correct? 1998

38. Thayer FJ. Herzog jc. Guttman 5D Strand spaces:Proving security protocols correct 1999(2-3)

39. Thayer FJ. Herzog Jc. Guttman JD Strand spaces:Honest ideals on strand spaces 1998

40. Schneider S Verifying authentication protocols with CSP 1997

41. Song D Athena:A new efficient automatic checker for security protocol analysis 1999

42. 郁志辉. 陈渝. 刘鹏 网格计算 2002

43. Xu Feng. Xie Li Service-Oriented Role-Based Access Control 2004(04)

44. Whitfield Diffie. Paul C. van Oorschot. Michael J.Wiener Authentication and Authenticated Key Exchanges 1992(02)

45. Syverson P. Meadows C A logical language for specifying cryptographic protocol requirements 1993

46. 卿斯汉 安全协议20年研究进展[期刊论文]-软件学报 2003(10)

47. Sandhu R. Conyne E J. Lfeinstein H L Role based access control models 1996(02)

48. Ferraiolo D F. Sandhu R. Guirila S. Kuhn D.R, Chandramouli R Proposed NIST standard for role-based access control 2001(03)

49. A Zieger Grid security:state of the art 2003

50. T Myer Grid watch:GGF and grid security vol 2004

51. Frank Siebenlist. Foster I Security for Grid Services 2003

52. SSL v3.0 specification

53. Dolev D. Yao A On the security of public key protocols 1983(02)

54. W Marrero. Ed Clarke. Somesh Jha Verifying security protocols with Brutus 2000(04)

相似文献(10条)

1. 期刊论文 李晶. 雷咏梅. Li Jing. Lei Yongmei 基于网格的3D Monte Carlo 算法及协同计算研究 -计算机应用与软件2006, 23(5)

Monte Carlo算法在高分子研究领域占有相当重要的地位. 本文在网格环境下实现了三维格点Monte Carlo算法在高分子链领域中的一个应用实例的并行研究了网格环境下的3D的MC算法协同计算, 并且实现了该算法的协同演示.

2. 期刊论文 赵辉. 李明楚. ZHAO Hui. LI Ming-chu 基于虚拟组织的网格安全需求分析模型 -计算机工程 2008, 34(24)

网格环境下多用户参与的协同计算是网格计算的重要应用方向. 网格计算的复杂性导致网格安全需求复杂. 该文提出一种基于虚拟组织的网格计算多用户协同关系描述模型, 在其基础上构建网格安全需求分析模型, 实现了网格环境下多用户协同计算的安全需求形式化描述, 把网格协同计算环境下的不同安全需求统一在同一种理论体系中.

3. 期刊论文 赵辉. 李明楚. 王智慧. ZHAO Hui. LI Ming-chu. WANG Zhi-hui 一种基于虚拟组织的网格安全协议形式化验证方法 -计算机工程与应用2007, 43(24)

虚拟组织是网格计算的基本管理单元, 而协同计算组是虚拟组织形成的基础. 对应于网格计算的复杂性, 网格安全协议的分析与证明十分复杂. 通过引入网格计算信道的概念, 在传统Strand Space理论的基础上提出了一种基于虚拟组织的网格安全协议形式化验证方法, 实现了网格环境下多用户协同计算安全协议的分析与证明.

4. 期刊论文 那丽春. 刘念祖. 徐伦彦. 俞时权. NA Li-chun. LIU Nian-zu. XU Lun-yan. YU Shi-quan 基于多智能体的多机群网格模型 -计算机工程与设计2007, 28(16)

在由多计算机机群构成的网格环境下, 为了实现数据并行型计算, 提出了一个基于多智能体机制的网格开发模型. 给出了由多计算机机群组成的网格、逻辑计算机机群、数据并行型计算和一系列Agent的定义. 利用管理智能体、独立计算智能体、协同计算智能体以及协同计算组之间的协同计算机制来实

现数据并行型计算.描述了网格计算过程.实践表明,该模型有效地适应了多机群网格环境的异构性、动态性等特性,提高了计算资源的利用率.该模型适合于基于网格的并行型计算.

5.期刊论文 袁秀梅.杨峰.徐志勇.Yuan Xiumei.Yang Feng.Xu Zhiyong 网格协同计算中任务调度系统研究 -北京工业职业技术学院学报2008,7(3)

网格环境下协同工作机制的研究具有重要的理论与应用价值,本文分析了网格环境下协同工作中任务调度的基本功能需求,并结合现有Globus平台对其中的关键技术进行了研究,给出了调度系统的原型设计.

6.会议论文 何建农 网格计算技术在土地信息管理中的应用 2006

网格计算技术在GIS领域的应用已成了当今GIS研究的前沿课题.针对土地信息管理的现状和问题,结合其应用特点和需求,设计了土地资源信息网格的组织框架,提出了采用网格GIS的技术解决方案,研制了应用实例,为土地信息管理的资源共享、信息服务和协同计算提供新的思路和有效的方法.

7.期刊论文 马晓宁.李明楚 一种改进的安全协议形式化需求语言 -电子技术应用2006,32(3)

对原有的安全协议形式化需求语言进行了改进,使其能适用于复杂的分布式系统.使用改进后的语言描述了网格环境下多用户协同计算中科学计算问题的安全需求.

8.学位论文 汪先明 网格协同计算中监控模型研究与系统设计实现 2009

网格监控为网格系统中其他网格中间件提供与资源有关的重要性能数据，供终端用户浏览决策提供数据，是网格系统进行资源发现、性能监控与调整、错误发现与纠正的依据，是保证资源得到最大最有效利用的保障，是保证作业任务顺利完成的重要支撑。在网格从基础理论研究阶段逐步走向初步应用阶段的今天，网格用户对网格监控与发现提出了新的要求，如用户需要了解其任务的执行情况以及对资源的使用情况等，同时网格计算面向的应用问题也是复杂多变的。

因此，本文针对网格中间件对监控这一块缺乏足够支持的现状，分析用户的监控对象与监控任务，分析实现任务监控要解决的关键问题，找到监控系统目前的难点与不足，指出监控系统的目标与结构，并给出了相应的解决策略。

接着，着重研究网格监控组件与监控模型，提供了实现监控模型的可行策略，剖析了WSRF框架下监控模型的组成结构、服务开发流程。研究了面向网格用户的监控服务开发的常用方案。

最后，应用和扩展MDS组件开发监控系统的开发方案，开发部署了一网格监控服务系统，这样就方便用户以web方式获取所需的监控信息，为网格体系中信息共享提供技术支持。

特别是对于监控系统也注重了它的安全性，从细粒度的信息安全到传输层、转换层进行了一些系列的配置与开发，这样使得监控系统本身这个层面就具有抵抗各种不安全因素的能力。

系统开发部署完成后，进行了各种测试。测试结果显示基本达到了预想的开发结果。

总体上，本文对网格监控模型进行了卓有成效的探索，对协同计算中监控系统的开发部署的各方面也进行了仔细的考虑，最后实现了设计初衷。

9.期刊论文 李明禄.薛广涛.刘飞.洪锋.Li Minglu.Xue Guangtao.Liu Fei.Hong Feng 基于多域和层次结构的网格数字图书馆 -现代图书情报技术2005,""(10)

数字图书馆的超大容量数据决定了现有的数字图书馆一定要采用分布式技术,资源的多样化特性决定了数字图书馆需要采用统一的平台来屏蔽资源的异构性.网格技术在以上两个方面都能满足数字图书馆的需求.网格技术与数字图书馆的结合是数字图书馆的发展趋势.本文在将数字图书馆和网格结合的基础上提出了基于多域和层次结构的网格图书馆.

10.期刊论文 陈亚玲.桂小林.王庆江.钱德沛 基于代理的网格计算中间件 -计算机研究与发展2003,40(12)

WADE系统是基于代理技术实现的一个可屏蔽异构和分布性的动态自适应的校园计算网格.提出了基于代理技术在校园网格内实现并行计算的方法,详细论述了基于代理的网格计算中间件的体系结构和主要模块功能,阐述了利用代理实现异构编译、协同计算的过程,给出了代理的Java实现方法.利用软件代理实现网格计算中间件,可以解决异构计算平台下多种并行编程环境的协同计算问题,为用户提供统一的服务接口,这将大大增强系统的可用性.