虚拟组织中的一种信任管理机制*)

王惠芳　郭　中　郭金庚　黄永忠　陈海勇
(解放军信息工程大学信息技术学院　郑州450002)

摘　要　灵活、安全地共享虚拟组织内的资源是虚拟组织的目的之一，也是需要研究的重要课题之一。本文提出了一种新的授权模型，即基于角色证书和策略的信任管理机制，来解决虚拟组织中的资源共享问题，并给出了原形系统。

关键词　虚拟组织，信任管理，分布式授权

A Trust Management Mechanism in Virtual Organization

WANG Hui-Fang　GUO Zhong　HUANG Yong-Zhong　GUO Jing-Geng　CHENG Hai-Yong
(Security Information School of PLA Information Engineering University, Zhengzhou 450002)

Abstract　To share resources of virtual organization(VO) flexibly and securely is a goal of virtual organization and an important task needed to research. This paper proposes a new authorization model, which is a trust management mechanism based on role certificate and policies, to resolve sharing resources problem in VO. And this paper describes a prototype implementation of this model.

Keywords　Virtual organization, Trust management, Distributed authorization

1　引言

在大型网络中连接有许多存储设备、高性能计算设备、特殊仪器等，它们属于不同的自治组织。在网络环境下，共享和合作使用这些设备和仪器，是进一步扩展网络应用范围的新方向。这些应用包括科学合作实验室，医疗研究(如基因)等为特定项目组成的团体。在团体中，为了实现共同的目标，团体中的成员或组织共享资源。这种资源共享不仅仅是文件传输，还包括对计算机、软件、数据、设备和其他资源的直接访问。这种共享应是高度可控制的，资源提供者和使用者都明确地、仔细地定义了哪些可以共享，谁可以共享，以及在什么条件下共享。

一组个体和(或)机构组成的正式和非正式的组织，在这个组织中的成员能在一定的规则下共享资源，称这种组织为虚拟团体或虚拟组织(virtual organization，VO)[5]。虚拟组织不同于传统的组织，它动态地聚集了同意共享资源的许多个体或机构，但它们仍旧属于不同的真实组织，仍然被它们所在的真实组织的内部规则和策略所控制。

由于 VO 中，各成员和组织之间的存在时间、拥有的权限和资源的数量、种类等都会动态地变化，这使得虚拟组织中的实体(包括用户、成员、资源、组织等)需要保持一种非常动态的共享关系。在 VO 中如何有效地管理这种共享关系是目前研究的一个重要课题。

本文提出了一种新的授权模型，即基于角色证书和策略的信任管理机制(RPTM)来解决 VO 中的资源共享问题。

本文第2部分分析和总结在 VO 中实现资源安全共享的要求。第3部分讨论 Globus 关于 VO 的授权机制。第4部分讨论我们开发的基于角色证书和策略的信任管理系统(RPTM)，以及如何支持 VO 中的资源共享。第5部分相关工作比较，及将来的工作。

2　VO 中资源安全共享的要求

传统的认证加 ACL 表的访问控制形式已经不再适合于 VO 中的资源访问控制，比如，在一个多机构的合作项目中得到一个共享计算资源的时间分配表。按照传统的访问控制模式—认证＋ACL 表，资源提供者就必须事先与成员建立直接的信任关系。这样合作机构中成员的每一次变动，项目负责人都要与资源的管理员联系，如给新的项目成员创建新的帐户。再者，当工程实施方案发生变化时，项目负责人不得不让资源管理员重新调整分配表，使得项目成员的权利和优先权与当前的合作相一致。

这种项目负责人与资源管理者的不断交互加重了资源管理者的负担，并且项目负责人管理的灵活性也大大降低，当项目的成员很多时，简直很难实现管理。

因此需要采用新的授权机制。在给出我们设计的授权机制之前，先分析 VO 对资源共享的要求：

可扩展性　管理用户的代价(如添加或删除用户)不应该随着加入 VO 的资源提供者的数量增加而增加。另外，对资源提供者来说，由于对于不同组织有不同的访问控制策略，资源管理的代价随着 VO 中的组织个数增加是合理的，但不能随着组织大小，或组织内的动态变化而增加。

灵活性和表达性　用分布方式实施 VO 中的协定和策略，一方面要具有灵活性，易于管理和实施。如果策略的微小改动，要涉及到大量的组织间交互，是不满足灵活性的。另一方面，策略语言能够正确地、明确地表达策略的含义，不存在二义性。即各个组织对策略的解释应该是一致的，这样才能正

*)该课题受到军队攀登工程项目资助，编号为00230。**王惠芳**　博士生，主要研究分布式系统的安全问题。**郭　中**　博士生，主要研究方向：分布式对象处理、多 Agent 系统。**郭金庚**　教授，博士生导师，主要研究方向：分布式对象处理、多 Agent 系统、信息安全。**黄永忠**　讲师，主要研究方向：分布式对象处理、软件工程。**陈海勇**　研究生，主要研究方向：分布式系统安全。

确地实施策略。

兼容性　在VO中，每个自治组织可能已存在授权机制。由于技术和实际的原因，这些已存在的安全框架不能在一夜之间取代，以实施单一的安全技术，采用单一的安全策略，因此要求能兼容已存在的授权机制。

表1　虚拟组织中的策略层次

一层	虚拟组织的整体策略	
二层	资源提供方的资源管理策略（内部策略和对外策略）	自治组织的策略（内部策略和对外策略）

策略分层　本文把VO中的策略分为二层，三部分，如表1。第一层虚拟组织的整体策略，它反映了VO中各个组织间的信任关系，以及共享资源的规则。第二层包括自治组织的策略，和资源提供方的资源管理策略，这两部分的策略又分别可以分为内部策略和对外策略。内部策略由于是各自内部的策略，因此尽可能地相互独立，策略的内部变化尽可能不影响其它策略，如自治组织内部策略的变化不会影响到资源管理者的策略。对外策略是组织间交互的策略，要能正确反映VO的整体策略。

提供何种资源共享机制才能满足上述要求，从而实现对VO中的资源共享进行有效、灵活地管理。这是目前需要研究的一个课题。

3　Globus中的授权机制

Globus中GSI(Grid Security Infrastructure)提供了让用户和应用程序安全访问资源的库函数和工具包。本文不详细介绍GSI，感兴趣的读者可以参看文[2]。本文只给出它的授权机制。

GSI支持本地策略本地实施的概念。为达到这个目的，GSI提供了转换GSI用户的身份(如用户身份证书中的唯一名字)为本地身份(如，Kerberos实体，或本地Unix用户帐户)的机制，如图1。转化后，本地身份能够用本地策略判定资源访问，如文件访问，计算时间等。显然这种名字映射表的形式可扩展性很差，当用户很多，范围很广，并且动态变化的情况下，管理员很难管理。尤其在用户与资源提供方事先没有信任关系的情况下，这个表格无法建立。

```
#GSI用户的身份                                                    本地身份
"/C=US/O=Globus/O=NPACI/OU=SDSC/CN=Rich Gallup"                   rpg
"/C=US/O=Globus/O=NPACI/OU=SDSC/CN=Richard Frost"                 frost
"/C=US/O=Globus/O=USC/OU=ISI/CN=-Carl Kesselman"                  u14543
"/C=US/O=Globus/O=ANL/OU=MCS/CN=Ian Foster"                       itf
```

图1　将GSI名字映射到本地名字的文件，该文件由Globus管理员维护

在引言中我们提到，VO中的成员是动态变化的，采用这种静态的名字映射表，不适合于VO中的资源动态共享。

基于上述情况，有一些研究团体开发了The CAS(团体授权服务)系统[5]集成到Globus中，我们将在第5部分将它与我们的工作进行比较。

4　基于角色证书和策略的信任管理系统(RPTM)

4.1　RPTM模型

见图2，RPTM的功能是将外域用户动态地映射到本域的帐户，或本域角色。本文将一个帐户名，或一个用户名统称为角色名称，实际上这是可行的，比如一个用户名可以看作只有一个成员的角色名。本文中域的概念为一个独立的自治组织，在它里面实施着单一的、一致的本地安全策略。

RPTM信任管理系统将推导资源提供者与未知的外域用户之间是否存在信任关系，从而决定是否按用户的请求授权。RPTM将根据用户申请的角色 R 和一组证书集合 C，以及本地策略 P，回答用户申请的角色与本地策略是否一致。RPTM可用 f 函数描述为 $f(R, C, P) \rightarrow$(Yes,No)。

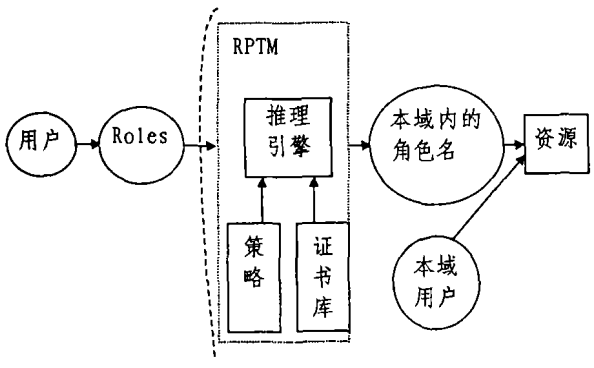

图2　RPTM将外域的用户动态地映射到本域的角色

4.2　RPTM的角色证书

角色证书的构造包括：角色，角色名称，实体。实体为代表用户的公钥，用 A, B 表示。因为角色可能属于不同的域空间，因此需要在描述时在角色名称前加一个限定，如 $A.r$ 表示 A 域的角色 r。在证书里的发行者隐含了角色属于哪个域。设 $Rmem(role)$ 表示角色 $role$ 的成员集合。角色证书分三种类型：

1. 将一个实体定义为一个角色。

$A.r \leftarrow B$　　(类型1)

这个证书表明 A 定义用户 B 为角色 r，设 $Rmem(A.r) = \{x \mid x \in A.r\}$，则 $B \in Rmem(A.r)$。

2. 基于角色的分布式授权

$A.r \leftarrow B.r_1$　　(类型2)

$A.r \leftarrow A.r_1.r_2$　　(类型3)

这两种证书表明将符号"←"前的角色权限授予符号"←"后的角色。

证书类型2的含义为：$Rmem(A.r) \supseteq Rmem(B.r_1)$。

证书类型3的含义为：A 定义角色 $A.r_1.r_2$ 具有角色 r 的许可权，即 $Rmem(A.r) \supseteq Rmem(A.r_1.r_2) = \bigcup_{B \in Rmem(A.r_1)} Rmem(B.r_2)$。如果 A, B 为不同的域，可以实现基于角色的分布式授权，即 B 定义的角色 r_2 的成员可以获得 A 定义的角色 r 的许可权。

如果 A, B 为同一个域，则可以实现本域中角色的分层。本系统限制了证书中角色的层次最多为两层，如果需要更多层，则可将多层分解。

RPTM的证书用 *s-express* 语言表达，可读性强。有四个域，*Issuer*，*Subject*，有效期，签名：

Issuer(发行者)：{*name*，"发行者公钥的 *hash* 值"：K_1，"角色名称"：*RoleName*}

注释："角色名称"域中的值是发行者定义的角色名；

Subject(主体):有三种形式:①{公钥的 *hash* 值:K_i} 表示实体 K_i 的属于 *RoleName*,即 $K_i \in Rmem(K_I \cdot RoleName)$;②{公钥 *hash* 值:$K_i$,角色名字 r} 表示该公钥实体 K_i 范围下的 r 角色成员属于"角色名称"域中的角色,即 $Rmem(K_i \cdot r) \subseteq Rmem(K_I \cdot RoleName)$,$K_i$ 与 K_I 相等时可以省略;③{角色名字 $r_1 \cdot r_2$},表示发行者实体 K_I 范围下的 $r_1 \cdot r_2$ 角色成员属于"角色名称"域中的角色,即 $Rmem(K_I \cdot r_1 \cdot r_2) \subseteq Rmem(K_I \cdot RoleName)$。

有效期:

签名:

注:证书主体的第一种形式 $K_i \in Rmem(K_I \cdot RoleName)$。当且仅当 $Rmem(K_I \cdot RoleName) = \{K_i\}$ 时,称为 K_i 与 $K_I \cdot RoleName$ 绑定。

下面举例说明,一个网上购书活动将 XXUniversity 的 student 角色成员定义为 discount 角色,从而可以购买折扣 70%的书。网站的公钥 hash 值为|+/94qiphSxJawXGbkcYh1A==|,证书有效期为不超过2003-10-01_00:00:00。

```
(cert
   (issuer
   (name (hash md5 |+/94qiphSxJawXGbkcYh1A==|) dis-
      count ))
   (subject (hash md5 |+/94qiphSxJawXGbkcYh1A==|)
      XXUniversity student)
   (not-after "2003-10-01_00:00:00"))
signature
```

例1(1) 证书(1)将角色 discount 的权限赋予 XXUniversity 的 student,即发行者|+/94qiphSxJawXGbkcYh1A==|定义 XXUniversity 的 student 角色可以获得 discount 角色的权限。

如果有证书(第一种类型):

```
(cert (issuer
      (name (hash md5 |+/94qiphSxJawXGbkcYh1A==|)
         XXUniversity ))
      (subject (hash md5 |Z5pxCD64YwgS1IY4Rh61oA==|))
      (not-after "2003-10-01_00:00:00"))
   signature
```

例1(2) 证书(2)说明对于发行者,XXUniversity 的公钥 hash 值为:|Z5pxCD64YwgS1IY4Rh61oA==|

则可推导出:

```
(cert (issuer
   (name (hash md5 |+/94qiphSxJawXGbkcYh1A==|) dis-
      count))
   (subject (hash md5 |Z5pxCD64YwgS1IY4Rh61oA==|) stu-
      dent)
   (not-after "2003-10-01_00:00:00"))
signature
```

例1(3) 证书(3),证书(3)与证书(1)的含义相同,但证书(1)的可读性强,易于理解。

4.3 策略语言

策略语言要能够正确、明确的表达授权者的意愿,不能有二义性。RPTM 的要求是易于理解,易于描述,能够根据角色证书和策略将实体动态地映射到角色。另外由于是开放分布式系统,部分策略需要发布,因此可读性要强。RPTM 采用 XML 语言描述策略。

RPTM 的策略主要是描述成为某格角色的信任要求。推理引擎需要根据用户请求的角色、提供的证书与本地的策略是否一致来判定是否授予该角色。

策略中,有一个特殊的角色"self",代表资源的提供者,是一个自签名的证书。每一个角色可以有多个规则(RULE)定义一个实体如何成为一个角色的成员。RULE 之间是或的关系。也就是说只要有一个 RULE 符合,则就可以将实体映射到一个角色。每一个 RULE 有多个约束条件,在 INCLUSION 标签中表示,INCLUSION 之间关系是与的关系,即所有约束条件都符合才能满足这个 RULE。

例2 映射 partnerhospital 角色的规则

```
〈?xml version="1.0" encoding="UTF-8"?〉
〈!DOCTYPE policy SYSTEM "Policy.dtd"〉
〈policy〉
   〈ROLE NAME ="self"/〉
   〈ROLE NAME=" partnerhospital"〉
  〈RULE〉
      〈INCLUSION FROM="self"〉〈/INCLUSION〉
 〈/RULE〉
 〈RULE〉
     〈 INCLUSION NAME = " hospital " FROM = "
     partnerhospital" THRESHOLD=2〉
     〈/INCLUSION〉
 〈/RULE〉
  〈/ROLE〉
〈/policy〉
```

例2表达的是如何将一个医院加入到"partnerhospital"的角色中,有两个 RULE,第一个 RULE 是拥有"self"签发的"partnerhospital"的角色证书;第二个 RULE 是拥有两个"partnerhospital"角色成员签发的"hospital"证书。

4.3.1 ROLE 标签 本策略主要由一组角色组成,每一个角色用〈ROLE〉标签定义。〈ROLE〉的唯一属性为名字,即角色的名字。在〈ROLE〉标签中有一个或多个〈RULE〉标签,每一个 RULE 定义成为该角色的约束条件,因此只要实体满足一条 RULE 约束条件,则可映射为该角色。

4.3.2 RULE 标签 一个 RULE 定义了成为一个角色的约束条件。这里的约束条件为必须的证书集合,这些证书集合能够证明发行者属于某个角色。如例2的第二 RULE 规定必须有两个(THREDHOLD=2)属于"partnerhospital"角色的成员颁发的"hospital"证书,才能映射为"partnerhospital"角色。

4.3.3 INCLUSION 标签 定义了必须存在的一个证书,或推导的证书。INCLUSION 标签包含的属性有:NAME,FROM,THRESHOLD,DEPTH,del-DEPTH。

例如〈INCLUSION NAME= "roleName" FROM="IssuerRole"〉〈/INCLUSION〉表明存在一个角色名为 roleName,发行者为角色 IssuerRole 的证书。属性 NAME 对这个证书定义的角色进行限制,如果没有 NAME 这个属性,则默认值为与角色名相同。属性 FROM 对证书的发行者进行限制,必须属于 IssuerRole 角色的一个成员。

INCLUSION 标签之间是与的关系,可以表达多种角色(属性)交集的约束关系。例如:

```
〈RULE〉
 〈INCLUSION NAME=" student " FROM=" University" 〉〈/IN-
    CLUSION〉
〈INCLUSION NAME=" member " FROM=" Volunteer"〉〈/IN-
    CLUSION〉
〈/RULE〉
```

即一个 INCLUSION 表示请求一个角色的实体应该具有的角色。设 RPTM 的推理引擎用映射函数 f 表达,则有下式成立:

$A.r = f(r_1 \cap r_2 \cdots \cap r_n)$($r_i$ 表示角色(属性),$A.r$ 表示映射的角色)

属性 THRESHOLD,它可以用来表达动态域结构,即要求某个角色的 k 个不同的成员签发的角色为属性 NAME 的角色证书。如例2的第二条 RULE 中 INCLUSION。要求两个具有属性"partnerhospital"(角色)的成员颁发的"hospital"证书,才能映射为"partnerhospital"角色。同样有下面的式子成立:

$A.r = f((\frac{k}{n} IssuerRole).roleName)$.

(*IssuerRole* 表示发行者角色，*roleName* 为证书定义的角色名，$A.r$ 表示映射的角色)

属性 DEPTH，表示该条件中，从 self 到 IssuerRole 的委托深度，它表达了对直接信任和推荐信任有不同的信任程度。默认值为无穷。比如说，A，B 是直接信任的 hospital 角色，C，D 为 A，B 共同推荐而成为 hospital 角色，如果 DEPTH 深度为2，则 self 不信任 C，D 为推荐者。同样有下面的式子成立：

$A.r = f(S_1, S_2, \cdots, S_k)(k \leqslant DEPTH, S_i$ 与 S_{i+1} 是委托与被委托的关系，$S_1 = self)$

属性 del-DEPTH，表示从 IssuerRole.roleName 到实体的委托深度，默认值为无穷。这个属性能够限制 IssuerRole 对角色 roleName 的委托深度，防止权限的无限委托。

4.4 资源提供方的设置

从图2，可以看出 VO 中的资源提供方需要安装 RPTM，设置资源访问控制策略和证书库。

下面举例说明，资源提供方如何与其他组织、域建立信任关系。

例3 域 B 与资源提供方 FileServer 协商使用某些资源的许可权，它将这些权限赋予角色 Programmer。策略描述如下所示：

```
〈?xml version="1.0" encoding="UTF-8"?〉
〈!DOCTYPE policy SYSTEM "Policy.dtd"〉
〈policy〉
〈ROLE NAME ="self"/〉
〈ROLE NAME = "ParaVO"〉
  〈RULE〉
    〈INCLUSION NAME= "ParaVO" FROM="self"〉〈/INCLU-
      SION 〉
  〈/RULE 〉
 〈/ROLE 〉
〈ROLE NAME ="Programmer"〉
  〈RULE〉
    〈INCLUSION FROM="ParaVO"〉〈/ INCLUSION 〉
  〈/RULE 〉
〈/ROLE 〉
〈/policy〉
```

例3中，第一个角色“ParaVO”的策略说明，只有拥有 FileServer 资源提供方签发的“ParaVO”角色证书的实体，才能映射到角色“ParaVO”。

第二个角色“Programmer”的策略说明，只有拥有 ParaVO 成员签发的“Programmer”的角色证书才能映射到角色“Programmer”。

假设 Kw，Kv 分别是文件系统 FileServer 和域 B 的公钥。FileServer 将颁发以下证书：

Kw.ParaVO <------- Kv

这个证书证明域 B 属于 FileServer 的 ParaVO 角色成员。通过这个证书和例3中 ParaVO 角色的策略说明，FileServer 建立了和域 B 的信任关系。

再通过例3中第二个角色“Programmer”的策略，说明拥有角色 ParaVO 颁发的 Programmer 角色证书的成员，属于 FileServer 的 Programmer 角色成员，即可以获得 FileServer 的 Programmer 角色的权限。通过这个证书和策略 FileServer 将角色 Programmer 的权限委托给域 B，即拥有域 B 颁发的 Programmer 角色证书的成员可以获得 FileServer 的 Programmer 角色的权限。

因为公钥不易记忆，为了使证书的可读性好，可颁发两个证书，如下：

Kw.ParaVO <-------域 B

Kw. 域 B <-------Kv

这两个证书合起来与上面第一个证书含义是相同的，因为 Kv 是域 B 的公钥，但由于公钥不便于记忆，阅读者不能直接从第一个证书看出谁属于 ParaVO 这个角色。可以看出后面两个证书的可读性好。

4.5 资源使用方的配置

域 B 加入 VO，要按照 VO 的策略与 VO 中的资源提供方协商，设置资源共享策略。仍以例3为例，则域 B 将要做以下工作。

域 B 安排本域内部的角色 C-Programmer 的所有成员可以获得 FileServer 的 Programmer 角色的权限。John 是角色 C-Programmer 的成员之一，将 John 映射到资源提供方 FileServer 的 Programmer 角色。假设 Kj 是 John 的公钥，则域 B 需要颁发以下证书：

(1) Kv. Programmer <------ C-Programmer (not-after "2003-10-01_00:00:00")

注意这个证书 name 后跟的公钥为域 B 的公钥 kv，说明发行者是域 B，是域 B 定义的 Programmer 角色证书，该证书证明角色 C-Programmer 的成员都属于角色 Programmer 的成员。这个证书是 RPTM 中一个特别的证书，在这个例子中，域 B 中角色 Programmer 仅仅是一个角色名字，不是真正的角色。这样资源提供者可以根据 VO 的策略来起角色名称，而不用关心域 B 内的角色或用户名。另外，由于域 B 面临的不止一个资源提供者，假设有 n 个资源提供者，域 B 有 m 个成员有权使用，则需要颁发 mn 个证书，而使用第二类型证书，只需 n 个第二类角色证书和 m 个角色证书。显然更易于管理。

证明 John 属于角色 C-Programmer 的证书，John 的公钥为 Kj：

(2) Kv.C-Programmer <-------Kj (not-after "2004-10-01_00:00:00")

可以看出证书(2)的有效期比证书(1)长，证书(1)的有效期与资源提供方的策略有关，证书(2)的有效期由域 B 的内部策略决定。

4.6 RPTM 的推理引擎

推理引擎也是 RPTM 中非常关键的一部分，它将根据本地策略，证书和用户请求的角色，推导出用户是否可以映射到该角色。由于目前策略语言是单调的(不包含否定语句)，该算法不处理证书吊销，过期的情况。也就是，假设提供给推理引擎的证书都是有效的。由于证书是分布存放的，让推理引擎到处查找是否存在某个证书是不现实的。我们让用户在发出请求时，输入证书存放的地址，由推理引擎远程获取。

RPTM 的推理引擎可用 f 函数描述为 f (R, C, P) --> (Yes, No)，输入参数 R 表示用户请求的角色，C 表示一组集合证书，P 表示策略。

下面仍以例3为例进行分析，域 B 的用户 John 申请 FileServer 的角色 Programmer。以例3的策略，FileServer 和域 B 提供的证书为参数，分析 John 是否能映射到 FileServer 的角色 Programmer。同样设 FileServer，域 B 和 John 的公钥分别为 Kw，Kv，Kj。

策略 P：角色 ParaVO 成员定义的角色 Programmer 的成员，可以映射到 FileServer 的角色 Programmer。

证书 C：

(下转第85页)

12 徐志伟,李晓林,等. 织女星信息网格的体系结构研究. 2002(39):943～947
13 Tuecks S,Czajkowski K. Grid Service Specification. www. gridforum. org,2002-10-04
14 Foster I,Kesselman C. The Physiology of the Grid:An Open services Architecture for Distributed Systems Integration. Open Grid Service Infrastructure WG,Global Grid Forum,2002-06-22
15 Box D,Ehnebuske D. Simple object access protocol(SOAP)1. 1. http://www. w3. org/TR/SOAP,May 2000
16 Christensen E,Curbera F,et al. Web Service description Language (WSDL) 1. 0. http://www-106. ib-m. com/developer Works/web/library/w-wsdl. html,2000-09-25
17 Ariba,Inc. ,International,Microsoft Corp. ,et al. UDDI Technical White Paper. http://www. hpmiddlewa-re. com/downloads/pdf/web_services_technicalwhite. pdf,2000-09-06
18 Leymann F. Web Service Flow Language(WSFL 1. 0). http:4. ibm. com/software/solutions/webservices/pdf/WSFL. pdf, May 2001
19 The Globus Project. Globus Tookit. http://www. globus. org/Tookit/. 2003-07
20 Global Grid Forum,Global Grid Forum Overview. http://www. gridforum. org.
21 刘鹏. 网格发展趋势. http://www. gridhome. com/grid/messages. htm

(上接第24页)

上一小节,域B颁发的证书(1),(2):

①说明了 John 属于域 B 的 Programmer 角色成员。

FileServer 颁发的证书:

ParaVO 〈---- 域 B;
域 B 〈---- Kv;
} ParaVO 〈---- Kv;

②FileServer 将域 B 定义为 ParaVO 的成员。

推导过程:

用 Rmem(role)表示某个角色的成员集合。

由①:John∈Rmem (域 B's Programmer)

由②:域 B∈Rmem(ParaVO)

∴ John∈Rmem (ParaVO's Programmer)

由策略 Rmem (ParaVO's Programmer) ⊆ Rmem (FileServer's Programmer)

∴ John∈Rmem(FileServer's Programmer)

因此 John 可以以 FileServer 的角色 Programmer 访问 FileServer 的资源。

5 相关工作及将来的工作

由第3部分分析,网格的安全基础框架(GSI)没有提供 VO 的分布式授权,提供了静态名字的对应关系,来实现本地策略本地实施。因此,有的研究团体开发了基于 GLOBUS 的团体授权服务系统 The CAS[5],该系统是基于能力证书的系统,CAS 维护本 VO 团体策略数据库和本 VO 的相关资源。在访问资源之前,用户要向 VO 的 CAS 申请能够完成一系列行为的能力证书,CAS 认证该用户,如果资源请求与本团体的策略一致,CAS 将授予合适的能力证书给该用户。该用户可以用这个能力证书提交给资源服务,资源服务认证正确,则资源服务提供用户能力证书上描述的权利。这个能力证书可以一直使用直到过期。

该模型中 CAS 成为资源访问控制的瓶颈。VO 的策略是由 CAS 来统一实施。另外是能力证书的有效期问题,由于该模型没有给出能力证书的吊销机制,有效期太长,可能使得退出的用户非法使用资源。有效期太短,用户要频繁地申请能力证书。但这种模型可以根据需求动态地分配资源,比较适合于用户量小、资源需要细粒度划分的服务。而 RPTM 模型适合于用户量大、资源提供者多且粒度大的资源。

当每一个角色只有一个成员时,RPTM 可以简化为名字的动态映射,参见第三部分 Globus 目前的资源安全共享机制。

我们开发的基于角色证书和策略的信任管理系统(RPTM)支持组织间的直接信任、推荐信任等多种信任关系,以及支持角色的多种约束关系。本文给出的例子只用到了组织间的直接信任关系,以及角色的简单映射。由于 RPTM 独立于应用程序,因此它不仅仅适合用于 VO,还适用于其他跨域的分布式应用环境,如 Web 服务。

目前,我们将 RPTM 用于分布式计算多 Agent 系统,该系统的主要目的是为了利用网络中的空闲计算资源,完成大计算量的计算任务。需要跨域访问不同域中空闲主机资源。

随着网络技术的发展,如 Agent 技术、网格技术,分布式系统逐渐向开放、大规模发展,开放分布式系统中的信任管理也逐渐成为分布式系统安全研究的热点。本文只实现了分布式计算中 VO 的信任管理,下一步想进一步应用到其他应用领域。

参 考 文 献

1 The Globus Project. 1997 http://www. globus. org/
2 Globus Security Policy and Implementation. 1997: http://www. globus. org/security/
3 Foster I,Kesselman C,Tuecke S. The Anatomy of the Grid:Enabling Scalable Virtual Organizations. International Journal of High Performance Computing Applications,2001,15(3):200～222
4 Blaze M, Feigenbaum J, Lacy J. Decentralized Trust Management. IEEE Conf. Security and Privacy, 1996, Oakland, California, USA. http://www. crypto. com/papers/policymaker. pdf
5 Pearlman L, Welch V, Foster I, Kesselman C, Tuecke S. A Community Authorization Service for Group Collaboration. In: Proc. of the IEEE 3rd Intl. Workshop on Policies for Distributed Systems and Networks, 2001
6 Sandhu R S, Coyne E J, Feinstein H L,Youman C E. Role-based access control models. IEEE Computer,1996,29(2):38～47
7 都志辉,陈渝,刘鹏. 网格计算. 清华大学出版社
8 Abadi M, Burrows M, Lampson B, Plotkin G. A calculus for access control in distributed systems. Transactions on Programming Languages and Systems,1993,15(4):706～734
9 Blaze M, Feigenbaum J, Ioannidis J, Keromytis A D. The KeyNote trust-management system, version 2. IETF RFC 2704, Sep. 1999
10 Blaze M, Feigenbaum J, Lacy J. Decentralized trust management. In:Proc. of the 1996 IEEE Symposium on Security and Privacy, IEEE Computer Society Press, May 1996. 164～173
11 Blaze M,et al. The Role of Trust Management in Distributed Systems Security. Secure Internet Programming: Security Issues for Mobile and Distributed Objects, Vitek and Jensen, Eds., 1999, Springer-Verlag. http://www. crypto. com/papers/trustmgt. pdf