基于主从样式的移动代理安全模型的研究与实现*)

穆 鸿 王汝传 张登银
(南京邮电学院计算机科学与技术系 南京210003)

Research and Implement of Master-Slave Pattern Based Security Model for Mobile Agent

MU Hong WANG Ru-Chuan ZHAN Deng-Yin
(Dept. of Computer Science and Technology, NJUPT, Nanjing, 210003)

Abstract In the ages of current network technology, as a new technology, the emergence of mobile agent shows us a wider and wider application prospect in the fields of network technology, such as Network Management, Intelligence Information Retrieval and Electronic Commence and so on. The security is the essence of mobile agent technology. In this paper we analyze the security issues and survey some representative approaches to solve the problem in mobile agent systems. We focus on the issues about the agent's security and propose a master-slave design pattern based security model for mobile agent called SMAP(Secure Mobile Agent Platform). In the end we give an application example to show how to use these.

Keywords Mobile agent, Security ,Design Pattern, Aglets

1 引言

随着 Internet/Intranet/Extranet 的迅速发展,网络的开放性、共享性和互联程度不断扩大,WWW 被广泛用作信息发布、电子商务和各种娱乐的业务平台,对应于此,出现了许多新的网络技术,移动代理技术便是其中的一种。

从广义上讲,移动代理可以是任何代表某个用户的程序段,它可以在计算机网络中漫游,代表用户在不同的节点上进行交互工作。它的应用范畴包括在线购物、设备的实时控制以及分布式科学计算等。在具体实现中,用户或代理控制中心向网络发出一个(或多个)移动代理,该代理按照特定的路线(用户预定义,或由它自己确定)在网络中漫游并与各服务器节点交互,完成任务后携带计算结果返回(初始用户或代理控制中心)。

移动代理具有很多优点,如减少网络流量、增加客户机和服务器的异步性、便于负载均衡和容错、支持移动客户和服务定制等。它在通信网路管理、智能网领域、Internet 上的智能信息检索以及分布计算领域等具有广泛的应用前景。

2 移动代理系统中的安全问题及其防护

我们将移动代理的安全分为三个方面:传统的安全即通信基础设施的安全,任何基于网络的应用将面临数据链路的不安全环节,移动代理也不例外,可以采用通用的保护通信通道安全的方法。目前有一种基于"加密通道-权限控制"的代理系统通信安全性实现方法。另两个方面包括保护主机和保护移动代理我们将在下面分别阐述。

2.1 保护主机的安全

对主机安全性的攻击可分为四个主要的范畴:(1)泄漏:被非授权方获取数据;(2)篡改:通过非授权方更新替换数据;(3)资源偷窃:通过非授权方使用设备;(4)故意破坏:恶意入侵主机数据或设备,无明显的利益目的。

目前针对上面问题的可以考虑的措施有:

(1)沙盒模型:运行环境对移动代理访问本地资源的权限加以限制,使得它就好像运行在一个特定的盒子中,只能对本地资源施加有限的影响,而不能操作本地文件系统。Java Applet 是一个典型的例子,它不能访问本地主机上的文件,除源主机外,不能与其它计算机建立网络连接。

(2)签名、认证、授权和资源分配:使用某种算法对移动代理进行数字签名,签名信息包括:移动代理的创建时间、创建者和发送者等。运行环境首先验证移动代理的身份,判断它是否是合法的。如果验证成功,则允许其登录并给其分配相应的系统资源和操作权限(可通过存取控制表 ACL 来描述,预先在目的主机上建立账号,并为该账号分配相应的资源和权限)。

(3)Proof-carrying code:该方法的核心思想是:代理运行环境提供一套安全规则,且能够验证某个移动代理是否有违规行为。移动代理只要遵守这一套规则,就能得以运行。

(4)代码检验:代码验证程序对移动代理进行检验,如果它携带了非法指令,将不被执行。该技术对于执行层脆弱,容易被运行的移动代理所破坏的情况比较实用。

(5)限制技术:用限制技术对具有持久生存能力的移动代理进行控制。限制技术包括时间限制、范围限制、复制限制,它们对于控制移动代理"到处乱走"很有效。但由于它不能预测到所有可能发生的事情,有时反而会干扰移动代理执行任务。

(6)核查记录:对移动代理的所有活动进行记录,当其遭受恶意攻击时,可以寻找"当事人"负责。移动代理到达一个多代理系统后,接收方会验证其"身份",监视器根据验证结果决定是否接收该代理,以及制定什么样的安全策略。移动代理执行前,有代码验证器对其进行验证;移动代理执行过程中,监视器执行安全策略监控代理对系统资源的访问及限制其运行时间;移动代理离开时携带新的加密数据,监视器限制移动代理的去向。

2.2 保护移动代理的安全

主要是保护移动代理免受潜在的恶意主机服务器和环境

*)本文得到国家自然科学基金(60173037和70271050)、江苏省自然科学基金(BK2001123)、国家高技术八六三项目(2002AA776030)、江苏省计算机信息处理重点实验室基金(kjs03061)资助。王汝传 教授,博士生导师,主要研究方向是计算机软件理论、计算机网络及信息安全、移动代理技术等。穆 鸿 硕士研究生,主要研究方向为计算机网络、移动代理和信息安全技术。张登银 副教授,主要研究方向为计算机网技术。

的攻击与保护主机技术相比，移动代理的保护还处于初步阶段，因为移动代理是由主机执行的，它不得不在主机环境中公开它的数据和代码，也就冒着被恶意主机篡改、扫描甚至是终止的危险了，这些显然给问题的解决带来了一定的难度，目前主要有两种方法：一是基于检测的安全性即根据移动代理的执行结果来检测、判断其是否受到攻击，破坏，称为事后检测。当然也有事前检测的方法即不让代理到不安全的执行环境中去。另外还有主动的保护措施：即通过某种事前方法来拒绝或减少伤害。与这些方面相关保护方法有：

(1)利用加密函数的计算：在移动代理中，并不是所有的代码和数据都是隐私，人们可能只对保护其中的关键数据和算法感兴趣。因此，只要保护移动代理中部分关键数据和算法即可，如对某个计算函数进行加密，使攻击者无法了解函数的内部逻辑。如：Sander 和 Tschudin 提出了一个加密函数计算 CEF(computing with encrypted function)方法。

(2)使模糊码(有限黑盒子法)：既然可对函数进行加密，那么应该可以对整个移动代理进行加密。因此在 CEF 的基础上，Stuttgart 大学的 Holy 提出了黑匣子的思想，用黑匣子来防御恶意主机对移动代理的攻击。其核心是从一个给定的代理规范来产生可执行的代理，并且产生的代理是不可被攻击和修改的。

其它方法还有部分结果封装(Partial Result Encapsulation)、共享路线记录(Mutual Itinerary Recording)、利用复制和投票的路线记录(Itinerary Recording with Replication and Voting)、执行追踪(Execution Tracing)以及 Enviromental Key Generation 等。

3 移动代理环境构建

我们构造的安全移动代理平台是基于 IBM 的移动代理环境 Aglets 的。Aglets 是 Java 对象，在网络上它可以从一台主机移动到另一台主机。也就是说，一个在一台主机上执行的 Aglet 可以突然中断执行而被派遣到远程主机再开始执行。当 Aglet 移动时它将携带程序代码以及它所运载的所有对象的状态。一个内建的安全机制使得接纳非信任 Aglets 是安全的。它的系统目标是：

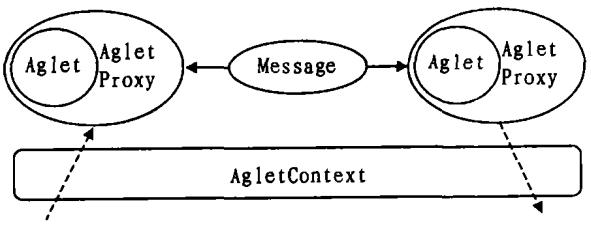


图1 Aglets 的 API 结构图

(1)提供一种简单而全面的模型，使得设计移动代理不需修改 Java 虚拟机或本地代码。

(2)支持动态和强大的通讯，允许代理像和已知的代理一样和未知代理通讯。

(3)设计一个可重用和可扩展的架构。

(4)用现有的 Web/Java 技术设计一个协调的架构。

(5)提供安全机制，它全面而简单，足够最终用户相信移动代理。

Aglet API 定义了移动代理的基本功能。图1给出了在 Aglet API 中定义的主要的接口和类以及这些接口之间的联系。而图2则给出了 Aglets 的对象生命周期图。我们可以看到 Aglet 并不直接与外界通信，它是经由一个称为 Aglet Proxy 的对象和外界打交道的，代理间的通信是以消息的方式进行的。Aglet 运行的环境成为 Aglet 上下文。在经过创建或克隆后 Aglet 便开始了它的生命周期，通过发送和召回，Aglet 实现了在不同上下文中移动，最后 Aglet 要么被释放掉要么被存储于媒质中，从而结束了自己的生命周期。

Aglets 架构包括两个层次和两种 API，这些 API 定义了访问它们功能的接口。

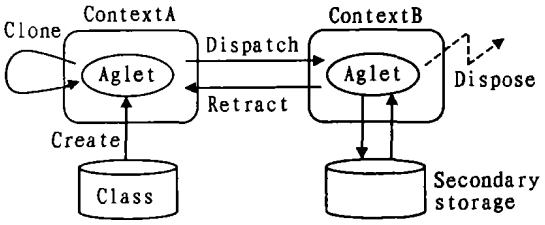


图2 Aglets 的生命周期图

Aglets 运行时层是 Aglet API 的实现，定义了 API 组件的行为像 AgletProxy 及 AgletContext。它提供了创建、管理及派遣 Aglet 到远程主机的基本功能。通讯层主要负责传输序列化了的代理到目的地以及召回它。它还支持代理到代理的通讯并使得代理管理简单化。Aglets 的架构图如图3所示。

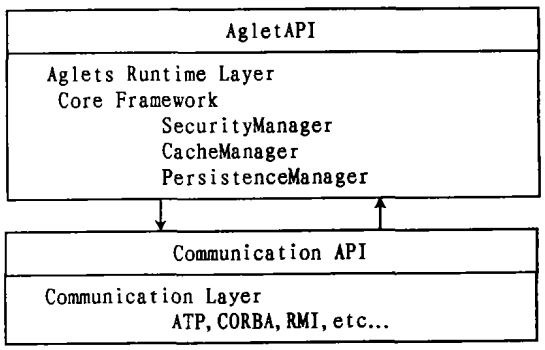


图3 Aglets 的系统架构图

Aglets 为开发移动代理应用提供了丰富的 API，可扩展的架构以及安全措施等，是开发移动代理应用的理想选择。更重要的是它提供了基于样式的丰富的 API，只要对平台进行适当的扩展就可以实现我们在前面所述的基于主从样式的安全移动代理应用方案。

4 安全模型

我们将主要精力集中在保护代理方面，已有的保护代理的思路很明确即采用检测或主动保护机制保证代理的安全，已有的采用检测的方法是基于主机可否信任而决定代理是否发往目的地，这必然限制代理的应用而主动保护方式又较复杂，其次这些方法不能将代理运载的敏感信息和一般功能(特定任务的)隔离开，也增加了代理被攻击及重要信息泄漏的危险。我们把两种方式集合起来提出基于主从(Master-Slave)设计样式的代理保护模型。

4.1 主从样式

主从样式是一种基本的样式，提供允许主 agent 把任务委派给从 agent，从 agent 移动到指定的目的地，完成指定任务后返回结果的机制。它包括两种类：主类及多个从类，从类

完成子任务(一个或多个),若需要则返回结果给主类,重要的是它们无需知道任务是如何划分以及整个任务最终是如何完成的。如图4所示。

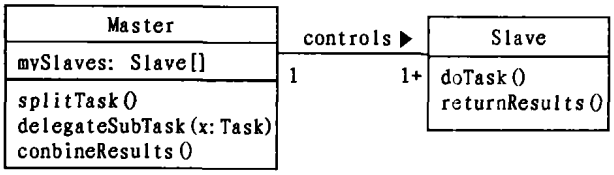


图4 主从样式

基于主从样式的移动代理的应用包括以下4个基本步骤:1)创建主代理;2)主代理创建从代理;3)主代理委派子任务给从代理;4)从代理返回结果后,主代理合并结果。

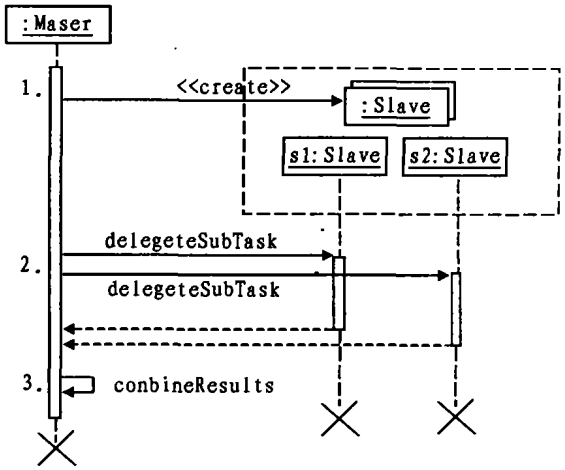


图5 主从样式应用步骤

我们用 UML 的序列图来描述这个步骤,如图5所示。主从样式的优点主要有以下几个方面:

(1)可扩展性,只要主从是松耦合的,改变从可以不需要改变主,甚至可以让不同的类来完成相同的任务。

(2)将计算及其它代码分离,因为可以将任务下放,并且可以将敏感信息保存在主代理中,所以从代理的计算可以在不涉及敏感信息的前提下独立完成。

(3)有效性,因为可以有多个从来并行完成任务,所以应用的有效性是显而易见的。

考虑安全性,我们希望代理不要传输敏感数据显然这是不现实的,那么可不可以借助主从样式来间接完成这个任务呢?答案是肯定的。方法是让主代理携带敏感数据停在信任的安全主机上,而产生的从代理只需携带足够完成子任务(由主代理分配)的数据即可。

4.2 控制模块

为了有效控制主代理和从代理之间的关系,我们在系统中引入控制模块来控制代理行为,我们定义以下四种控制类型,我们以伪代码给出了一些实例。

(1)任务分解:用于决定如何产生从代理,是并行还是串行、是否按功能分等。

```
c.setTaskPolicy(1)//1为方式选择字表示串行方式
//产生从代理
proxy=context.createAglet(null,"msagent.Worker",getProxy();
c.SetAccessAddress(proxy,"url1,url2");//设置将要访问的地址
proxy.dispatch(url1)//先访问地址1到达后再访问地址2
c. setTaskPolicy(2)//2为方式选择字表示并行方式
//产生从代理
proxy=context.createAglet(null,"msagent.Worker",getProxy();
sproxy=context.createAglet(null,"msagent.Worker",getProxy();
c.setAccessAddress(proxy,"url1");//访问控制对象的方法,设置将要访问的地址
c.setAccessAddress(sproxy,"url2");//访问控制对象的方法,设置将要访问的地址
proxy.dispatch(url1)//访问地址1
proxy.dispatch(url1)//访问地址2
```

(2)路由控制:决定路由选择方式,最简单是顺序访问还是按可用性访问。

```
c.setRoutePolicy=seq|default //seq 表示顺序而 default 表示按可用性访问
```

(3)执行检查:用于检查代理系统是否可以执行代理。

```
if !(c.checkEnv())then exit
```

(4)合并/任务结果:决定结果的处理方式以及下一步行动,如:在并行方式下可能是选择价格最底的供应商再派交易代理前往交易。

```
doMergeAction(){
if minPrice then
{
proxy=context.createAglet(null,"msagent.Buger",getProxy();
proxy.dispatch(url);
}
```

4.3 安全移动代理平台(SMAP)

SMAP 是在 Aglets 的基础上建立起来的,这样可以充分利用 Aglets 给我们提供的丰富的 API,包括代理的创建、移动、销毁等原语操作,以及安全措施和良好的可重用的设计样式。SMAP 的基本结构框图如图6所示,我们在基本 Aglets 平台的基础上对其扩展部分进行增强同时增加了一个控制模块 CM。CM 模块主要用于配合主从样式工作的。如主代理将按照任务分解策略决定如何产生从代理,我们在控制模块部分已讲述过相关内容这里不再赘述。

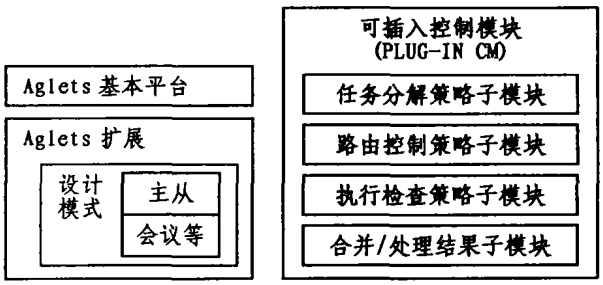
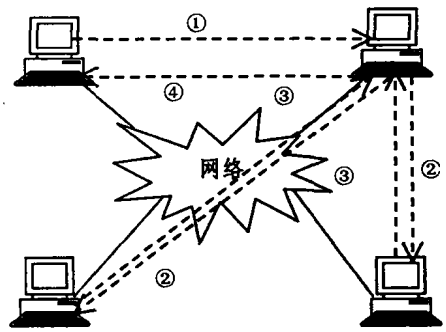


图6 SMAP 结构框图



①代理发起主机生成并发送主代理到信息主机;②主代理依照控制模块的要求分解任务,确定路由并生成和发出从代理;③从代理完成各自任务后返回到信息主机;④要么进行其它任务,要么主代理携带处理结果返回到代理发起主机。

图7 基本运行过程

利用以上的 SMAP 开发的移动代理应用的工作流程如图7所示。我们必须在主代理出发前定义好以上四个策略,在实现上就是给定四个方法的具体实现,我们对特定模块的公共功能进行了抽象,但依照应用不同,对具体策略实施是不尽

相同的。图7描述了一个大概的工作流程，没有涉及细节问题。在随后的部分我们以开发的一个简单实例来描述这种框架的具体应用。

5 应用实例分析

系统的构建基于以下一个场景：客户要到北京出差，他想知道各个航空公司所提供的飞机票的价格和公司名，并购买最低价格的商家的票，同时他决定预定客房也是最低价格的客房。传统的实现方式是：客户登录各个公司的商务网站，分别查询符合条件的飞机票的价格。在对各个公司的价格都查询完毕再决定自己购买哪个公司的飞机票，定房也是如此。在基于移动代理的系统中，我们可以指定各个供应商的地址（事实上，这个地址集合可以通过查询商务中心来定，在演示中我们指定），然后派出主代理到其中一个信任主机上（通过查询主机信任度表决定如图8所示），在信任主机上再产生查询价格的子代理到各个主机上（安全性的考虑），这些子代理可以是并行的即有几个子代理去并行查询各个主机，也可以是串行的即子代理在一个主机上查询完毕再到另一个主机这样最简单，我们只需一个子代理就可以完成任务了。在系统中我们选择了按功能划分的并行方式，因为这样可以直观地考查主从代理是如何工作的。相关的角色以及控制策略如表1和表2所示。

TrustTB：表

Host	Vaight
atp://192.168.0.1:3000	0
atp://192.168.0.1:4434	1
atp://192.168.0.2:4434	2
atp://192.168.0.2:3000	3
atp://192.168.0.3:4434	4
atp://192.168.0.3:3000	5
atp://192.168.0.1:6000	7
atp://192.168.0.1:5000	8

图8 主机信息度表

表1 角色

商务代理	主代理
购票代理	从代理一
定房代理	从代理二

表2 控制策略

路由	顺序
执行/检查	无
合并/处理结果	价格最低
任务分解	按功能分

我们以在4434端口作为源宿主地并且在地址选择框中输入下列两个地址为例：atp：//192.168.0.2：3000；atp：//192.168.0.1：5000来说明系统工作的流程。当然也可以其它地址输入，但要注意输入的地址的格式。同时，要在数据表中建立相应的主机的项。基本流程是：主代理知道192.168.0.1：5000是最信任主机决定到那儿去然后我们根据功能生成一个购票代理一个定房代理，购票代理在最便宜的地方定票而定房也是在最便宜的地方实现的，为简单起见我们让主机加端口号来区分不同主机，并且每个主机承担了票和房的双重身份。实验表明这种应用是切实有效的。

结束语 解决移动代理的安全问题是使其真正实用的关键要素。本文在阐述移动代理安全问题及其对策的基础上从设计模式的角度来研究移动代理本身的安全，提出了安全的移动代理平台（SMAP），并给出了一个应用实例，下一步我们将把传统的安全技术和基于设计配置的安全技术结合起来以期建立真正安全实用的移动代理应用系统。

参 考 文 献

1 Chess D. Security Issues in Mobile Code Systems. In：Giovanni Vigna(ED.)，Mobile Agent Security，LNCS 1419，Springer，1998. 1～14

2 Karnik N M，Tripathi A R. Design Issues in Mobile Agent Programming Systems. IEEE Concurrency，1998，6(3)：52～61

3 Wilhelm U G. et al. A Pessimistic Approach to Trust in Mobile Agent Platforms. IEEE COMPUTING，September·October 2000. 40～48

4 Schaaf M，Maure F. Integrating Java and CORBA：A Programmer's Perspective. IEEE INTERNET COMPUTING，January-February 2001. 72～78

5 Wang X F，Yi X ，Lam K Y. Secure Information Agent for Internet Trading [A]. 11th Australian Joint Conference on Artificial Intelligence'98，Vol 1 544[C]. Brisban，Australia：Springer-Verlag Publishers，1998. 183～194

6 Farmer W M，Gutman J D，Swarup V. Security for Mobile Agents：Authentication and State Appraisal. In：Proc. Of the 4th European Symp. On Research in Computer Security，Rome，Italy，LNCS No. 1146，Springer，1996. 118～130

7 Eckel B. Thinking in Java，2nd Edition，Release 11. Prentice-Hall，2000

8 Pfleeger C P. Security in Computing，Second Edition. Upper Saddle River，New Jersey：Prentice Hall，1997

9 Schneier B. Applied Cryptography：Protocols，Algorithms，and Source Code in C. IS John Wiley & Sons，ISBN：0-47-111709-9

（上接第61页）

$|\Sigma|$），类进行初始化可以在匹配前完成，因此这点时间在整个分析还原过程中可以忽视不计。M 等于 len，在中西文混合串中，Σ 取值范围为0x00..0xff，因此 $|\Sigma|=256$。为了存放 MT，需 m 个字的附加空间，另外引入字符数组 inter_mt [256]，将 Pat 中的 $p_1\cdots p_m$ 映射到1…m，而将非 Pat 中的所有字符映射到 inter_mt 占的空间为256/4＝64，因此总的附加空间为64＋m。

串匹配时间仅与文本长度有关，而与模式及文本内容无关，其最好、最坏及平均匹配时间均为 o(n)。

结束语 网络内容是计算机安全领域的又一个研究方向。本文提供高速网络环境下分布式网络审计系统对此进行了大胆的探索与实践，并且取得了较好的效果。系统提出的大流量数据分流概念、缓存管理技术、多字符集字符串匹配算法等在网络审计、入侵监测等领域都有重要的参考价值。

参 考 文 献

1 Fisk M，Varghese G. Fast Content-Based Packet Handling For Intrusion Detection. http：//www. public. lanl. gov/mfisk/papers/ucsd-tr-cs2001-0670. pdf

2 Desai N. Increasing Performance in High Speed NIDS. http：// www. linuxsecurity. com/articles/ intrusion_detection_article-4617. html

3 Baeya_Yates R，Gonnel G H. A new approach to text searching. Comm. ACM，1992，35(10)：74～82

4 邹旭楷，王素琴. 一种快速(汉字)字符串匹配算法. 小型微型计算机系统，1993(11)