移动 Agent 系统的安全性研究*

李 新, 吕 建, 曹 春, 冯新宇, 陶先平

(南京大学 计算机软件新技术国家重点实验室,江苏 南京 210093);
(南京大学 计算机软件研究所,江苏 南京 210093)
E-mail: lixin@softlab.nju.edu.cn; lj@nju.edu.cn
http://moon.nju.edu.cn

摘要: 移动 Agent 系统中的安全问题是阻碍其广泛应用的原因之一.通过对 Agent 的安全问题及其研究现状的分析,提出了解决一些安全问题的方法,并在自行研制的移动 Agent 系统 Mogent 中实现,对防止和解决移动 Agent 系统中的安全问题有一定的作用.

关 键 词: Agent;安全;访问控制;授权;电子货币
中图法分类号: TP309 文献标识码: A

1993 年,General Magic 公司推出了 Telescript,这是一种可以用来编写可移动代码的程序设计语言和环境,虽然在商业运作上没有取得成功,但从此在工业界和学术界掀起了对移动代码的研究热潮.1996 年,Java 语言的诞生和新型浏览器的出现,使 Applet 得到了广泛的应用,进一步促进了对移动代码的研究.

Robert S. Gray 给移动 Agent 下的定义[1]是:移动 Agent 是一个程序,该程序可以在网络中的主机上移动,并且在这些主机上执行.由此定义可以推知移动 Agent 具有以下一些特性:移动性、自主性、安全性、协同性、智能性和社会性等,这种新的计算方式与传统的 RPC 相比,具有以下优点[2]:

(1) 减轻了网络负载;(2) 对网络的延时和抖动不敏感;(3) 可以封装专有协议,在不兼容系统之间协调和互操作;(4) 异步操作不需要持续的网络连接;(5) 容易支持服务的个性化;(6) 更容易适应多变的网络环境.

虽然移动 Agent 技术具有很多优点,但是一个严峻的问题——安全却阻碍了它的应用.移动 Agent 可以把多台分布的计算机连接起来,构成一个计算基础设施,在其上可同时运行属于不同用户而且是潜在的不可信任用户的分布式应用程序.这些计算机分别属于不同的组织,有各自不同的用途,通过公共通信设施进行通信.在这样的环境下,存在着各种可能的不安全因素以及安全攻击.例如,未被授权的用户可以监听网络线路,在移动 Agent 传输过程中窃听甚至修改其代码或数据;当 Agent 运行时,可能会攻击当地的主机,故意占用过多资源,或者利用系统中的漏洞取得特权,进而攻击主机或其他 Agent 等等.

本文就移动 Agent 环境中的安全问题进行了一些探讨,第 1 节分析移动 Agent 环境中的安全问题和目前的研究现状.第 2 节着重介绍我们设计并实现的 Mogent 系统中的安全设施.最后是总结,并对今后的研究工作提出一些展望.

* 收稿日期: 2000-11-13; 修改日期: 2001-03-16
基金项目: 国家自然科学基金资助项目(69873021);国家 863 高科技发展计划资助项目(863-02-ZT02-01-4);国家杰出青年基金资助项目(61525204);江苏省应用基础基金资助项目(13J99016).

作者简介: 李新(1974－),男,江苏镇江人,硕士生,主要研究领域为对象技术,移动 Agent 技术;吕建(1960－),男,博士,教授,博士生导师,主要研究领域为软件自动化与形式化方法,移动 Agent 技术;曹春(1978－),男,江苏无锡人,硕士生,主要研究领域为对象技术,移动 Agent 技术;冯新宇(1978－),男,山东济宁人,硕士生,主要研究领域为对象技术,移动 Agent 技术;陶先平(1970－),男,博士,副教授,主要研究领域为对象技术,移动 Agent 技术.

1 移动 Agent 环境中的安全问题

许多研究移动 Agent 安全性问题的学者倾向于从组成 Agent 系统的各个部分来分析其安全问题.到目前为止,已经提出了不少的移动 Agent 模型.实际上,对于讨论安全问题,一个简单的模型就足够了.在一个移动 Agent 系统中有两个最重要的部分,一个是可移动的 Agent,另一个是支持 Agent 移动和运行的主机,有时又称平台或节点,Agent 由代码和状态两部分组成.最初用户发出 Agent 的平台称为初始平台(home platform),在移动 Agent 环境下,初始平台是最可信的,而且也是惟一可信的.

Robert S. Gray 把移动 Agent 环境中的安全性问题分为 4 类[1]:保护主机、保护其他 Agent、保护 Agent 和保护主机构成的网络.

1.1 移动Agent环境中的4类安全问题

1.1.1 保护主机

这类安全问题主要是恶意的或者是有错误的 Agent 利用主机上安全设施的不足或缺陷发起针对主机的攻击,主要有伪装、拒绝服务和未授权访问.

1.1.2 保护其他 Agent

这类问题是指 Agent 可能会利用系统的缺陷对在主机上运行的其他 Agent 进行攻击,包括伪装、未授权访问、拒绝服务和抵赖.其实这个问题可以看成是保护主机的子问题,首先,因为主机上的 Agent 支持环境可能就包含一些 Agent,如负责通信的 Agent,管理 Agent 名字空间的 Agent 等,对这类 Agent 的攻击等于是对系统进行攻击.其次,可以把主机上运行的 Agent 看成是主机的一部分资源,Agent 对其他 Agent 进行攻击也就是对主机的攻击.再次,从目前已有的安全措施来看,保护主机和保护其他 Agent 所使用的技术十分类似.

1.1.3 保护 Agent

在移动 Agent 环境下,不仅可能会有恶意的 Agent,而且也会存在恶意的主机对 Agent 进行破坏.由于 Agent 发送到主机后要在其上运行,它的代码和数据以及运行时刻的通信对主机来说都是暴露无疑的,可以说主机为刀俎,Agent 为鱼肉,主机可以对 Agent 采取任何动作,所以保护 Agent 是所有问题中最困难的.也正因如此,才吸引了许多人花费大量的精力进行研究.

1.1.4 保护网络

Agent 可以在一个站点上只消耗少量的资源,其行为也完全符合站点的安全策略,但它却暗中以隐蔽的方式破坏着网络的可用性,进而会使一些主机瘫痪.对这类攻击的防治也比较困难,从单个主机着手显然不能解决问题,必须从网络整体考虑.当网络从属于一个管理者时,网络上的各个主机之间比较信任,找到一个统一的方法相对来说还容易一些,否则,在像 Internet 这样无中心、无权威的网络上,困难就会大得多.

1.2 目前的研究现状

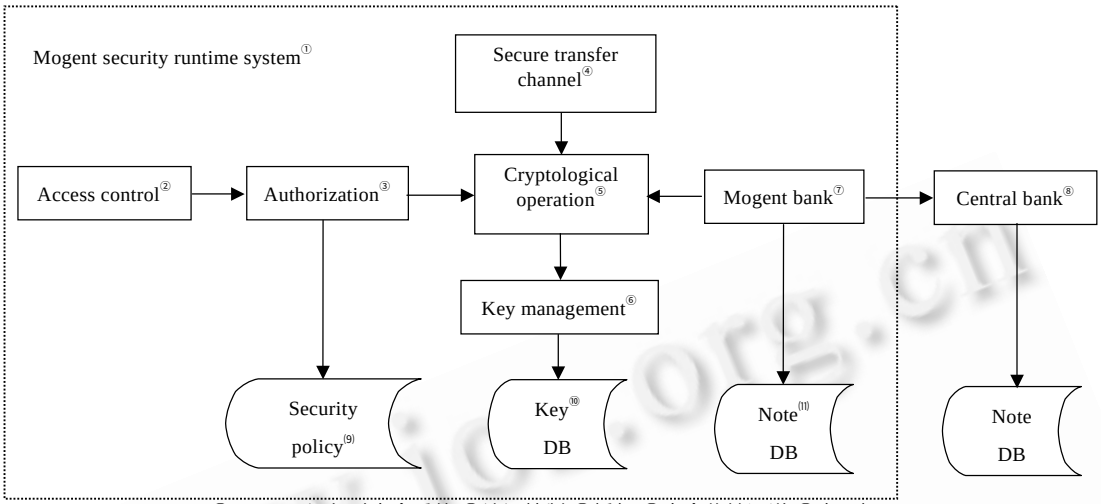
在移动 Agent 这一概念提出后不久,人们就意识到其中存在的安全问题并开始着手研究、借鉴现代分布式系统中的许多技术和方法,根据移动 Agent 的特点,进行改进和扩充,取得了一定的成果,提出了许多提高移动 Agent 系统安全性的方法,如基于软件的错误隔离[3]、解释执行代码、带数字签名的代码、状态评估[4]、携带证明的代码[5]、相互记录旅行历史[6]等都较好地解决了保护主机和保护其他 Agent 的问题,其中一些已经应用到实验系统中去,但是对于保护 Agent、保护网络目前仍无有效的方法.

2 Mogent 系统安全部分的设计和实现

Mogent(mobile agent)是南京大学计算机软件新技术国家重点实验室研制的基于 Java 的移动 Agent 系统,目前的版本是 2.0,运行在 Sun Solaris+JDK1.2.2 环境下,运行系统主要由 3 部分组成:迁移(包括 Mogent 的执行)[7]、通信[8]和安全.针对移动 Agent 环境中的 4 类安全问题,在已有研究成果的基础上,我们在 Mogent 系统中设计了安全总体结构,采取了一些防范措施,旨在有效地保护主机和网络,其中基于信任传递的授权模型和在 Agent 系统中使用电子货币的尝试具有一定特点.

2.1 Mogent安全子系统的总体结构

Mogent 中安全子系统由 6 部分组成,另外还有一些辅助设施,总体结构如图 1 所示.



①Mogent 运行时安全系统,②访问控制,③授权,④安全传输通道,⑤密写操作,⑥密钥管理,⑦Mogent 银行,⑧中央银行,⑨安全策略,⑩密钥库,⑪货币库.

Fig.1 Structure of security subsystem in Mogent

图1 Mogent 安全子系统结构

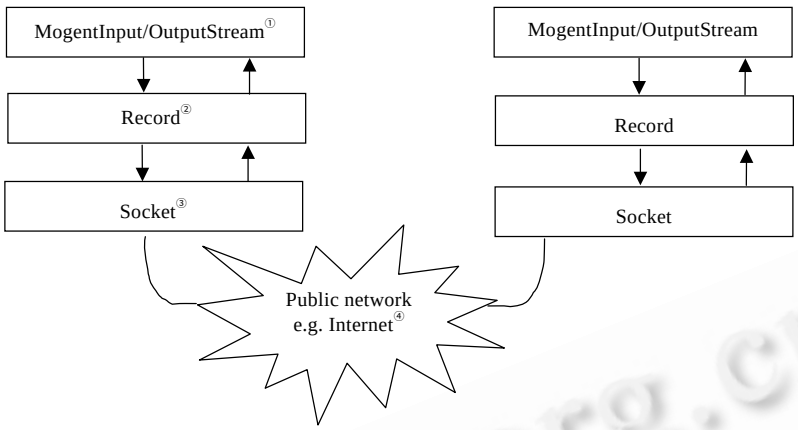
整个安全子系统的中心是密写操作和密钥管理器,因为在安全子系统中有大量的数据加密、解密、签名和验证操作,分布在其他各个部分.安全通道给 Mogent 传输和 Mogent 之间的消息传递提供保密的信道,授权根据用户的身份决定 Mogent 在运行时刻访问资源的权力,访问控制就是实施这些访问资源时限制的.Mogent 银行负责 Mogent 平台上电子货币的使用,它有两个作用:(1) 当外来的 Mogent 访问资源后,要向银行交费;(2) 平台上的用户发送 Mogent 时从银行中取出电子货币给 Mogent.电子货币的中央发行银行负责电子货币的发行、转账等业务.下面将对每个部分作详细介绍.

2.2 安全传输通道

Internet 早期的设计目标是使人们可以方便地进行网络互联,因此没有过多地考虑安全因素,这种情况一直延续到了现在.Internet 先天的在安全方面的缺陷给攻击者带来了极大的便利.Mogent 系统可能受到的攻击之一就是网络监听:因为网络上传递的信息绝大多数都是明文,攻击者可以轻易地窃取网络上传输的数据流并进行分析,得到传输的内容,甚至可以将内容修改后再回放到网络上去.Mogent 系统可能受到的攻击之二是欺骗:当一个 Mogent 系统正在与另一个系统交互时,对方实际上并不是自己想要交互的系统,而是通过某种手段进行了欺骗.

Netscape 公司推出的安全套接字 SSL[9](secure socket layer)技术综合运用现代密码学技术,可以在不安全的网络传输层服务之上建立一个可靠的安全通道.但由于在系统设计的时候尚未有 SSL 协议的 Java 实现,所以我们无法直接在 Mogent 系统中使用 SSL,因此根据 SSL 的原理和 Mogent 系统的要求,我们实现了一个稍微简单的安全传输通道,同时将对通道的输入和输出操作用专门的类封装起来,这样就会对上层透明.安全传输通道的结构如图 2 所示.

整个安全通道可分为 3 层:输入/输出流层、记录层和套接字层.最高层是输入/输出流层,它是通道和高层软件的界面,接收要传送的数据,用下面将提及的握手协议中协商的密钥进行加密,然后传递给下面的记录层,或者从记录层接收加密了的数据,进行解密,然后交给上层.由于会话过程中使用的加密算法是分组算法,因此加密过的数据是有边界的,而且解密时必须保持原边界,记录层就是用来管理加密数据的边界的.它从上层的输出流得到已经加密的数据,在其前面加上数据类型和长度,然后通过下层的 BSD Socket 将带边界信息的数据发送出去,或者从 Socket 读取数据,根据其中的边界信息,给上层的输入流读取整个加密数据分组.



①输入输出流层,②记录层,③Socket 层,④传输网络.

Fig.2 Structure of secure transfer channel

图2 安全传输通道结构示意图

当安全通道建立时,客户要对服务器进行验证,然后进行会话密钥的协商.在验证过程中,服务器要对客户端发送过来的具有一定随机性质的数据进行变换后再数字签名,将签名和自己的证书发送给客户端.客户端验证证书的正确性、签名的正确性和服务器名称是否与证书中的主题一致,如果全部通过验证,在会话密钥协商过程中,双方将协商在以后的通信过程中要使用的加密算法、密钥和可能的初始向量.一旦协商结束,后续的数据传输都是用会话密钥进行加密的.

2.3 授 权

Mogent 服务器收到一个 Mogent 后,要在它运行之前进行授权,即确定该 Mogent 可以访问哪些资源.首先,服务器通过 Mogent 所携带的数字签名确认其用户的身份,然后从管理者配置的安全策略中,检索出该用户的 Mogent 拥有的权力,授予这个 Mogent.资源访问控制部分将利用这些信息在运行时刻决定 Mogent 是否可以访问某资源.

2.3.1 授权模型

上面已经提及,授权是给一个用户的 Mogent 设置可访问的资源,在许多移动 Agent 系统中,其根据实际上是 Mogent 平台的管理者和 Mogent 用户之间的信任关系.如果 Mogent 平台管理者对 Mogent 用户比较信任,他认为 Mogent 不会有恶意的行为,就会分配较大的权力;反之,如果 Mogent 用户是一个不可信的甚至是匿名的,那么 Mogent 平台管理者只能分配很少的权力给 Mogent,以确保自身的安全.

我们认为,这种类似传统操作系统中授权的方法忽视了 Mogent 的流动性,尚有不足.Mogent 的一次旅行可能会经历多台主机,从 Mogent 服务器管理者的角度来讲涉及到 3 类用户,一是 Mogent 的发送者,二是管理者自身,三是 Mogent 旅行中其他服务器上的用户.上面的模型仅考虑了前两类用户,忽视了第 3 类.例如,在一个 Mogent 的旅行中,它从最初用户的计算机出发,经历了多台中间的 Mogent 服务器 A~H,最后到达服务器 I,那么服务器 I 上的管理者在给 Mogent 授权时不仅要考虑自己和最初发送 Mogent 的用户之间的信任关系,实际上还要考虑他和中间所有 Mogent 服务器 A,B,…,H 上用户的信任关系.由于目前安全技术尚存在许多不足,特别是针对恶意的主机还没有保护 Mogent 的有效方法,所以即使是一个信任的用户发出的 Mogent,在经过中途一个恶意站点时仍有可能被篡改,变成一个恶意的 Mogent,所以在授权时必须考虑所有上述 3 类用户.

有鉴于此,服务器管理者在配置安全策略时不仅要设置每一个用户的 Mogent 可以访问哪些资源,而且要对每一项资源设置可访问的信任主机,即确认 Mogent 访问过的服务器必须全部属于这些列出的信任主机时才能访问该资源,否则不可以.这种新的基于用户信任关系和访问历史的授权模型将给 Mogent 服务器管理者以更加细致的安全控制.

2.3.2 部分防篡改的路径历史记录

由于授权时使用到了 Mogent 访问过的主机信息,所以这些数据必须予以保护,防止被篡改,否则,恶意的用户可以通过修改路径信息隐匿掉 Mogent 曾经访问过恶意主机的事实,从而获得较大的资源访问权力.在恶意主机之间没有协作的前提下,可以通过互相协作的 Agent 来记录旅行历史的方法[6],但是需要一个额外的 Agent,效率不高.我们对此进行了改进,采用了在安全技术中常用的联锁(interlock)机制,在同一前提下可以防止对历史路径信息的篡改.

具体的做法是:每个 Mogent 携带一张历史路径(path history)表,由许多相同的表项构成,结构如图 3 所示.

Local node address①	Next node address②	Digital signature by local node③

①本节点地址,②下一节点地址,③本节点产生的数字签名.

(a)

202.119.36.173	202.119.36.174	…
202.119.36.174	202.119.36.175	…
202.119.36.175	202.119.36.176	…
…	…	…

(b)

Fig.3 Item of path history table and one instance of the table

图3 历史路径表项和一个历史路径表实例

每一表项中包含本节点地址和下一个节点的地址,还有本节点产生的数字签名,这样就把相邻的两个表项“联锁”起来,难以篡改.

用户在发出 Mogent 时,在路径历史表中增加第 1 条记录,填上“本节点地址”和“下一节点地址”,并进行签名.接收的服务器在对 Mogent 授权之前,要对路径历史记录进行检查,防止被篡改.检查时,对表中的所有表项逐一从上到下检查数字签名,并且任意相邻的两个表项必须一致,即前一表项的“下一节点地址”必须和下一表项的“本节点地址”相同.如有数字签名不正确或者相邻表项不一致就认为被篡改了,那么将给该 Mogent 最小的权力.在 Mogent 运行结束后发送至下一节点之前,在路径历史表中追加一条在本节点上访问的记录.

2.4 访问控制

访问控制的功能就是根据授权信息实时地监视 Mogent 对系统资源的访问,判断访问是否允许,如果可以就继续执行,否则掷出异常,终止对资源的访问.除此之外,访问控制还负责对 Mogent 资源访问的计费任务.在访问之前,要根据 Mogent 现有的货币量和待访问资源的价格判断它是否有支付能力,如果有,就把将要访问的资源的信息记入 Mogent 的账单,在 Mogent 运行结束后一起结算.

2.4.1 访问控制的实现

Java 运行系统对访问控制有很好的支持,通过安全管理器、访问控制器的设施,可以在 Mogent 访问受系统保护的资源时截获访问请求,然后检查该 Mogent 是否有权力访问.

Java 运行系统的访问控制还有许多不足,目前还不能对系统资源进行广泛而细致的管理,例如无法控制 Mogent 对 CPU 和内存资源的使用.JRes 项目组[10]在 Java 的 bytecode 中增加了对 CPU、内存等资源进行记账和控制的代码,在修改过的 Java 虚拟机上可以对系统资源进行更严格的控制,但却失去了 Java 平台无关的特性.

2.4.2 资源访问计费

对资源访问计费的工作是在安全管理器中完成的.当安全管理器检查当前线程是否具有资源访问权力的时候,首先调用访问控制器的 checkPermission 方法,如果允许访问,再判断该资源是系统代码自身访问的,还是在 Mogent 代码中访问的,或是为 Mogent 提供服务的 Mogent 系统代码访问的.判断方法是从顶至底分析方法调用栈,若没有一个类是 Mogent 类或其子类,则资源访问请求是系统发出的;若至少有一个方法是 Mogent 类或其子类的,则(a) 若从该类到栈顶至少有一个类是 mogent2 包中的(Mogent 系统的类都在 mogent2 包中)且是 PrivilegedAction 或 PrivilegedExceptionAction 的实例,则该资源访问请求是 Mogent 调用系统的服务而间接发出的;(b) 若从该类到栈顶没有一个类是 mogent2 包中的且是 PrivilegedAction 或 PrivilegedExceptionAction 的实

例,则该资源访问请求是 Mogent 直接发出的.目前,我们对后两种情况实行收费制度.先取得 Mogent 的账单和钱包,再从安全策略配置信息中检索出资源的价格,看看 Mogent 是否有支付能力.如果没有,Mogent 还是不能访问资源,抛出安全异常;如果有能力,安全管理器就将访问的资源和价格记入到 Mogent 的账单中去.

2.5 电子货币

在 Mogent 2.0 中开始使用电子货币,Mogent 要付费才能访问资源.目前,电子货币的使用处于初步尝试阶段,与真实的货币之间没有任何联系.我们认为,在 Mogent 系统中使用电子货币有如下 3 点优势:

(1) 本文的开始部分提到过移动 Agent 具有许多传统网络计算模型不具备的优点,很有可能成为一种新的计算方式.特别是随着 Internet 的迅猛发展,很多学者和工业界人士对移动 Agent 在电子商务中应用的前景非常看好.移动 Agent 如果能够使用电子货币和电子支付系统,将对其在电子商务中的应用大有益处.

(2) 由于运行和使用移动 Agent 系统有一定的开销和安全风险,所以很多用户可能不会接收外来用户的 Agent 访问自己的计算机系统.电子货币的使用将促使更多的人开放自己的计算机系统给其他用户使用,促进资源共享.例如,像密码破译这样计算量特别庞大的任务,以前常见的做法是在 Internet 上招募志愿者参加,每人计算一小部分.如果采用付费移动 Agent 的话,将会有更多的人乐意参加,从而更快更好地完成大计算量的任务.

(3) 可以有效地抵御拒绝服务攻击.拒绝服务攻击技术一般不是很先进,但往往很有效,原因主要有两点:一是攻击者可以以接近零的代价占用被攻击者大量的资源;二是被攻击者无法通过零资源消耗来抵御攻击.如果移动 Agent 需要付费才能访问系统资源的话,第 1 个条件不再满足,因此 Agent 很难发起有效的拒绝服务攻击.

实际的电子货币系统是十分复杂的,涉及的不仅有技术,还牵涉到金融制度、人们的社会关系等非技术因素.在 Mogent 系统中,我们主要关注基础技术,在电子货币的发行、Mogent 使用货币的方法和 Mogent 服务器与货币发行银行之间的交易协议方面作一些初步尝试.

2.5.1 电子货币的发行

由于真正意义上的电子货币具有不可追踪(匿名)、客户无关和非联机等特性,技术十分复杂[11],我们采用了一种简单易行的方式.Mogent 系统中的电子货币更类似于电子支票,包括以下几个部分:发行者、持有者、发行日期、货币的序号、面值和发行者对上述数据的数字签名,防止伪造和非法篡改.

发行银行确保其发行的每一货币序号的惟一性,并记录下每一货币的使用情况,防止两次和多次使用(double spending).为简单起见,Mogent 系统目前只支持一家货币发行银行.

2.5.2 Mogent 使用货币的方法

为了实现 Mogent 付费访问资源,系统中增加了不少部件:每个 Mogent 服务器上增加了 MogentBank,它负责接收外来 Mogent 的付费,同时给往外发送的 Mogent 提供电子货币;每个 Mogent 中增加了钱包 MogentWallet 和账单 Bill 对象,MogentWallet 是存放 Mogent 携带的电子货币的,Bill 记录在旅行中访问的资源和价格.

当用户生成一个新的 Mogent 时,同时也生成 MogentWallet 和 Bill 对象,内容为空.发送前,用户从本地的 MogentBank 中取出一些电子货币存入钱包中.当 Mogent 在执行过程中访问了收费资源后,Mogent 服务器把访问的资源和价格记录在账单上.执行结束后,从 Mogent 的钱包中取出一定量的货币,转给中央发行银行要求转账.中央发行银行在验证了货币的合法性后,生成新的货币给 Mogent 服务器,必要时给 Mogent 找零.

2.5.3 Mogent 服务器和电子货币发行银行之间的交易协议

Mogent 服务器在接收到 Mogent 的付款后需要将这些货币转成自己的,必须通过中央货币发行银行进行转账,这就涉及到 Mogent 服务器和货币发行银行之间的交易协议.我们参考 NetCash[12]中的交易协议,根据 Mogent 系统的需要设计了一个简易协议,在 Mogent 服务器和中央货币发行银行之间进行交易.图 4 是整个交易过程的示意,其中的④、⑦是在 Mogent 服务器上进行的,无须通过网络,而⑤、⑥是通过网络进行的,因此我们使用第 2.2 节中的安全传输通道,实现对中央银行的验证和交易过程的保密,但目前没有实现交易的匿名性(由于电子货币本身不具备匿名性)和不可抵赖性,还需进一步完善.

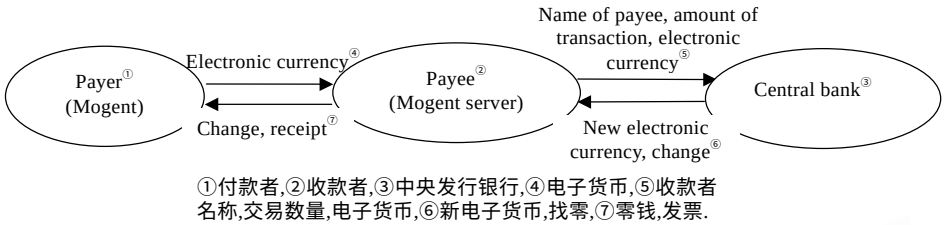


Fig.4 Payment protocol between Mogent server and central bank

图 4 Mogent 服务器和电子货币中央发行银行之间的交易协议示意图

2.6 密钥管理和密写操作

Mogent 中涉及了大量的密写操作,加密、解密、消息摘要、数字签名和认证以及与这些操作相关的密钥和证书.Java 及 JCE(Java cryptography extension)虽然有较强的支持,但仍存在以下不足:

(1) 由于美国密码软件产品受到出口的限制,Java 运行环境目前只支持 DSA 算法密钥和其证书的管理,不支持 RSA 密钥;而 DSA 算法只能用于签名,不能用于加密,功能不够.

(2) 密钥的存储缺省采用专有格式,结构不公开,特别是不能导出私有密钥.

鉴于以上两点,我们以自由软件 SSLeay 为基础,结合 Mogent 系统的需要,遵从工业界标准,进行密钥管理和密写操作,以期达到下列目的:

(1) 安全、可靠,所有私密钥必须用可靠的加密算法加密后存储,这是最基本的要求.

(2) 可以和其他应用程序安全地共享密钥、证书信息,给用户提供方便,无须为不同应用程序生成不同的密钥.

(3) 支持 RSA 密钥的管理.RSA 算法是目前应用得最为广泛的公开密钥算法,支持数据加密.

目前不提供公共密钥基础设施(public key infrastructure,简称 PKI),应当具备的密钥和证书生存周期管理的所有功能,如分发、取消、更新和归档等,目前这些工作手工完成.

2.6.1 密钥管理

Mogent 系统共涉及以下几种密钥和证书:

(1) CA(certificate authority)的证书;

(2) 用户 RSA 私密钥和证书;

(3) 服务器(包括 Mogent 服务器和中央银行服务器)RSA 私密钥和证书;

(4) 安全传输通道中的会话密钥,

其中会话密钥是由协商生成的,用毕即丢弃,因此不存在长期管理的问题,下面对前 3 种不同的密钥分别介绍其管理.

认证中心 CA 是为公共密钥颁发证书的机构,处于整个 PKI 的核心地位,是数字世界中信任关系的基石,一般由公众比较信任的银行、政府或独立机构设立,如中央银行、VeriSign Inc. USA 等,本身并不属于 Mogent 系统.但是为了实验方便,我们自己建立了一个认证服务中心,服务器软件选用 Netscape Certificate Server Version 1.01 for Solaris2.4, 2.5,可以接受证书请求,经审核后为个人用户和服务器颁发证书.整个 Mogent 系统中使用的证书,包括个人用户证书、Mogent 服务器证书和电子银行服务器证书都是由该认证中心颁发的.CA 对自己的公钥也颁发证书,即 CA 的证书,每个 Mogent 平台都存储着 CA 的证书,用来验证服务器和用户的证书.

为了和其他应用程序安全地共享用户密钥数据,Mogent 系统本身不产生用户密钥,而是从其他应用程序中导出(export),我们选用 Netscape 公司的 Communicator 浏览器.个人用户可以使用 Communicator 生成 512 位 RSA 密钥对,并向 CA 申请公钥证书.在 CA 颁发证书后,用户可以将私密钥和证书(链)按照 PKCS#12[13]中规定的 PFX 个人信息交换格式导出到文件中(导出的数据是加密的),然后其中的私密钥用 SSLeay 转储为加密的 PEM[14](privacy enhanced mail)格式,单独存为一个文件,同时 PFX 文件经 Windows 运行库将证书导出到单独的证书文件中去.

Mogent服务器使用的512位RSA密钥对由SSLeay生成,其中私密钥用DES算法经CBC模式加密,以PEM格式存储;公共密钥用SSLeay生成PKCS#10格式[15]的证书请求(certificate request),将证书请求发送给CA服务器请求颁发服务器类型的证书,证书主体(subject)的CN(common name)为服务器名称.证书颁发后,以X.509V3标准格式存储.

2.6.2 密写操作

Mogent中用到了对称和非对称密钥算法.对称密钥算法为DES,使用JCE1.2中的实现,全部采用比较安全的CBC模式,密钥长度为56位,可以对任何长度的数据进行加、解密.

非对称密钥算法为RSA,密钥长度为512位,是我们根据RSA算法自己实现了加密和解密过程.从RSA密钥中提取两个参数:模 n 和指数 e,对数据 x 的加密过程是[16]:

$$y=x^e \bmod n, 0\le y<n .$$

相应地,对加密数据 y 的解密过程为

$$x=y^e \bmod n, 0\le x<n.$$

注:指数 e 对于公共密钥和私有密钥是不同的.对数据的签名和验证操作类似,这里不再赘述,可参阅文献[16].

在对数据进行数字签名时用到的消息摘要算法,我们采用了MD5[17]un JDK1.2.2的实现,对任意长度的输入得到128位的摘要信息.

3 总结和进一步的工作

现在已经有了不少基于Java的移动Agent系统,总的看来,Mogent系统中的安全设施和其他基于Java的移动Agent系统大体相当,特别是在电子货币使用、基于旅行历史记录的授权模型等方面有一些自己的特点,但是所使用的一些加密算法或者已经过时,或者强度不够,容易攻破,今后将采取更好的方法.表1是Mogent与这些系统在安全方面主要特性的比较.

Table 1 Comparison of Mogent and other Java-based mobile Agent systems in security aspect

表1 Mogent与其他基于Java的移动Agent系统在安全方面的比较

Product①	Authentication②	Access control③	Transmission integrity④	Storage protection⑤	Digitally signed code⑥
Aglets	Domain authentication through symmetric algorithms⑦	Customizable through GUI⑧	Symmetric algorithms to detect potential tampering of incoming MAs⑨	No	No
Concordia	User and group identity encrypted with symmetric algorithm⑩	Permissions are stored in a preference file⑪	SSL protocol⑫	Yes, with symmetric algorithms⑬	No
Grasshopper	Yes, with X.509 certificates⑭	GUI-customizable security preference. The rasshopper security manager can be subclassed⑮	SSL protocol	No	Yes
Odyssey	No	No	No	No	No
Voyager	No	By using or subclassing the VoyagerSecurityManager class⑯	No	No	No
Mogent	Yes, with X.509 certificates⑰	Customizable through GUI⑱	SSL-Like protocol⑲	No	Yes

①系统名称,②验证方式,③访问控制,④传输一致性,⑤存储保护,⑥数字签名代码,⑦通过对称算法完成域验证,⑧通过图形用户界面配置,⑨使用对称加密算法检测迁入Agent是否被窜改,⑩用户和组标识使用对称算法加密,⑪使用配置文件存储权限控制列表,⑫SSL协议,⑬对称加密算法,⑭使用X.509证书,⑮通过图形用户界面设置,并且Grasshopper Security Manager可以被继承,⑯通过使用或者继承VoyagerSecurityManager类,⑰使用X.509证书,⑱通过图形用户界面配置,⑲类SSL协议.

虽然学者们对移动Agent环境中的安全问题进行了多年的研究,取得了许多成果,提出了不少针对具体问

题的方法,特别是对于保护主机问题解决得较好,但还有一些真正困难的问题尚未解决,例如移动条件下的数字签名和代码与数据的保护、拒绝服务攻击的防范等,目前还没有明显的进展,有待进一步深入的研究.

References:

[1] Gray, R.S., Cybenko, G., Kotz, D., *et al*. D'Agents: security in a multiple-language, mobile-agent system. In: Vigna, G., ed. Mobile Agents and Security. Lecture Notes in Computer Science 1419, Berlin: Springer-Verlag, 1998. 154～187.

[2] Chess, D., Harrison, C., Kershenbaum, A. Mobile agents: are they a good idea? 1994. http://www.research.ibm.com/iagents/paps/mobile_idea.pdf.

[3] Wahbe, R., Lucco, S., Anderson, T.E., *et al*. Efficient software-based fault isolation. ACM SIGOPS Operating Systems Review, 1993,27(5):203～216.

[4] Farmer, W., Guttman, J., Swarup, V. Security for mobile agents: authentication and state appraisal. In: Bertino, E., ed. Proceedings of the Computer Security——ESORICS'96. Lecture Notes in Computer Science 1146, 1996. 118～130.

[5] Necula, G.C., Safe, P.L. Untrusted agents using proof-carrying code. In: Vigna, G., ed. Mobile Agents and Security. Lecture Notes in Computer Science 1419, Berlin: Springer-Verlag, 1998. 61～91.

[6] Roth, V. Secure recording of itineraries through cooperating agents. 1998. http://cuiwww.unige.ch/~ecoopws/ws98/papers/vroth98c.ps.

[7] Tao, Xian-ping, Lü, Jian, Zhang, Guan-qun, *et al*. Design and implementation of a mobile agent structured migration mechanism. Journal of Software, 2000,11(7):918～923 (in Chinese).

[8] Tao, Xian-ping, Feng, Xin-yu, Li, Xin, *et al*. Communication mechanism in mogent system. Journal of Software, 2000,11(8):1060～1065 (in Chinese).

[9] Freier, A.O., Karlton, P., Kocher, P.C. The SSL protocol. 1996. http://home.netscape.com/eng/ssl3/ssl-toc.html.

[10] Czajkowski, G., von Eicken, T. JRes: a resource accounting interface for Java. ACM SIGPLAN Notices, 1998,33(10):21～35.

[11] Chaum, D., Fiat, A., Naor, N. Untraceable electronic cash. In: Goldwasser, S., ed. Advances in Cryptology——CRYPTO'88. Lecture Notes in Computer Science 403, Berlin: Springer-Verlag, 1990. 319～327.

[12] Medvinsky, G., Neuman, B.C. Netcash: a design for practical electronic currency on the internet. In: ACM SIGSAC. 1993. 102～106.

[13] RSA Laboratories, RSA Security Inc. Public-Key cryptography standards #12: personal information exchange syntax standard. 1997. http://www.rsasecurity.com/rsalabs/pkcs.

[14] Linn, J., *et al*. Privacy enhancement for internet electronic mail: Part I~Part IV. RFC1421-RFC1424, 1993.

[15] RSA Laboratories, RSA Security Inc. Public-Key cryptography standards #10: certification request syntax standard. 1993. http://www.rsasecurity.com/rsalabs/pkcs.

[16] RSA Laboratories, RSA Security Inc. Public-Key cryptography standards #1: RSA encryption standard. 1993. http://www.rsasecurity.com/rsalabs/pkcs.

[17] Rivest, R.L. The MD5 message-digest algorithm. RFC 1321, 1992.

附中文参考文献:

[7] 陶先平,吕建,张冠群,等.一种移动 agent 结构化迁移机制的设计和实现.软件学报,2000,11(7):918～923.

[8] 陶先平,冯新宇,李新,等.Mogent 系统的通信机制.软件学报,2000,11(8):1060～1065.

Research on Security in Mobile Agent System*

LI Xin, LÜ Jian, CAO Chun, FENG Xin-yu, TAO Xian-ping

(*State Key Laboratory for Novel Software Technology*, *Nanjing University*, *Nanjing* 210093, *China*);

(*Institute of Computer Software*, *Nanjing University*, *Nanjing* 210093, *China*)

E-mail: lixin@softlab.nju.edu.cn; lj@nju.edu.cn

http://moon.nju.edu.cn

Abstract: Security is an important obstacle to the application of mobile agent technology. After analyzing the security problems and current status of research work, several methods are proposed to solve some of the problems, and are implemented in the mobile agent system Mogent. These methods can prevent or solve some of the security problems.

Key words: Agent; security; access control; authorization; electronic currency

* Received November 13, 2000; accepted March 16, 2001

Supported by the National Natural Science Foundation of China under Grant No.69873021; the National High Technology Development 863 Program of China under Grant No.863-02-ZT02-01-4; the National Natural Science Foundation for Excellent Young Scientists of China under Grant No.61525204; the Science Foundation for Applied-Fundamental Research of Jiangsu Province of China under Grant No.13J99016

2003 年中国智能自动化会议(CIAC 2003)

征 文 通 知

由中国自动化学会智能自动化专业委员会主办，香港中文大学承办，中国人工智能学会、IEEE 控制系统学会北京分会、IEEE 控制系统,机器人与自动化香港联合分会、香港城市大学、香港理工大学协办的 2003 年中国智能自动化会议将于 2003 年 12 月 15 日~17 日在香港中文大学召开。

征文内容:

智能控制及自动化;人工智能及应用;人工神经网络;模糊控制与系统;进化计算及应用;智能机器人技术;计算智能及软计算;智能传感器及应用;智能制造系统;离散事件系统;混合系统及应用;多智能体系统;混沌、分形与小波;信息处理;智能管理与决策;智能设计与制造;智能建模与仿真;故障诊断;通信网络技术;智能交通系统;人机交互技术;数据挖掘;虚拟现实技术;多媒体技术;计算机视觉;模式识别;图像处理技术;其他相关领域。

论文要求:

在国内外杂志或会议上未曾发表过的论文;篇幅不超过 A4 纸 6 页。加附 200 字作者简介。

关键日期:

2003 年 6 月 15 日前投送符合清稿要求的全文一式 2 份或 WORD 电子稿。

2003 年 8 月 15 日前发出录用通知。

会议联系人:

钱宗华,清华大学计算机科学与技术系 100084

010-6278-8939 (O); 010-6278-4458 (H)

E-mail: hqzh@s1000e.cs.tsinghua.edu.cn

孟庆虎,E-mail: ciac2003@acae.cuhk.edu.hk

会议网站:

http://zncaa.gongkong.com