第 33 卷 第 9 期 计 算 机 工 程 2007 年 5 月
Vol.33 No.9 Computer Engineering May 2007
·安全技术· 文章编号：1000—3428(2007)09—0163—02 文献标识码：A 中图分类号：TP311

基于移动 Agent 的协同电子商务安全认证机制

周克江
(湖南省第一师范学校信息技术系，长沙 410002)

摘 要：如何提供一个协同、开放、灵活并高效的安全服务，是协同电子商务必须解决的一个重要问题，移动 Agent 为解决这一问题提供了一个新的方法。该文介绍一种基于历史事件的安全认证机制，以期进一步完善协同电子商务中的安全服务问题。
关键词：移动 Agent；协同电子商务；数字证书；PSL；安全策略

Mobile Agent-based Cooperative E-commerce Security Authentication Mechanism

ZHOU Kejiang
(Dept. of Information Technology, Hunan No.1 Normal College, Changsha 410002)

【Abstract】It is an important problem that providing a concurrent opening flexible and high effective security service for the cooperative e-commerce. A novel method is provided by mobile agent to resolve this problem. A history events-based security authentication method is given by the paper for resolving some security service problems in the cooperative e-commerce.
【Key words】Mobile agent; Cooperative e-commerce; Figure certification; Security policy language(PSL); Security strategy

1 概述

随着电子商务的发展经历了以 IT 厂商和媒体为主的第 1 阶段和以电子商务服务商为主体的第 2 阶段后，现在正在进入以传统企业为主体的第 3 阶段——协同电子商务阶段。协同电子商务要求在多个参与角色之间协调多项流程，它的特点和要求决定了它必须有一个强大的、安全的网络平台作为支撑，以保证其内部和外部复杂的交互和协作要求。

电子商务的这种新的发展趋势，是企业自身发展的需要，同时也表明了企业商务应用正向着开放性、协作性和服务化方向发展，而已有安全解决方案很难为企业间的电子商务交易提供协同的、开放的安全服务。实际上，电子商务交易就是通过数据的安全传输实现对关键资源的安全访问和操作，而认证和授权是保证交易安全性的主要技术，其中，认证是确定交易双方(即资源访问者)合法身份的过程，授权则是对合法用户分配资源访问许可权的过程，二者相结合，共同保证合法用户对资源的正常访问，尤其是可以限制用户对关键资源的访问，防止非法用户的侵入或者因合法用户的不慎操作所造成的破坏。目前常用的认证技术包括：口令认证，SSL，数字证书(如X.509 证书)，Kerberos，Smart Cards以及基于SAML认证等[1]，常用的授权模型及实现机制主要有：以资源为中心的访问控制列表(ACLs)，以用户为中心的能力凭证(Capability)式以及以代码为中心的Java授权模型(JAm)等[2~5]。在实际应用中，企业根据自身的需求，自主地选择安全技术和实现方案，因此，在跨企业的电子商务交易中，由于其安全技术与实现方式的多样性，以及安全信息描述手段的多样性，导致安全系统大多只能以企业为边界提供安全服务，很难与其他企业的安全系统进行互操作，有关交易和用户的信息无法从一个站点带到下一个站点，不仅造成交易执行过程中用户需要进行多次认证，企业间的业务合作也很难实现。因此，如何提高跨企业边界的安全信息互操作性，使得交易企业可从其他企业获取有关用户和交易等授权参考数据，是目前国际工业界和学术界所致力解决的关键问题之一，然而由于缺乏统一的数据表示和交互规范，这方面的工作进展不大。为了促进上述问题的解决，本文提出一种基于移动Agent的协同电子商务安全认证机制，以推动协同电子商务安全服务的发展。

2 移动 Agent 的安全性

移动Agent就其本质是指:在计算机网络中代替用户应用的一段程序。能自治而主动地从网络的一台主机迁移到另一台主机，代表用户进行相关的计算，完成特定的功能，并向用户返回相应的计算结果。移动Agent在其应用中有着明显的优势：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，支特移动客户和服务定制等。其应用领域有：分布式系统管理，网络信息搜索，电子商务等[6]。

尽管移动 Agent 有着广泛的应用前景。国外一些公司和高校研究机构推出自已的移动 Agent 系统。但就目前的一些应用而言还不太成熟，其中的主要制约因素之一便是移动 Agent 安全性。移动 Agent 因其程序的自由迁移和应用的灵活性而产生许多安全性问题。主要有以下两方面：(1)恶意 Agent 对目标主机的成助。当一个非可靠移动 Agent 迁移到一台主机后，可能窃取主机的敏感信息，耗尽主机的资源，发起拒绝服务(DoS)以拒绝给其它 Agent 提供服务，甚至传播病毒或破坏整个主机系统等。(2)任意的目标主机或代理服务器对移动 Agent 的伤害：当移动 Agent 迁移到一个不可信任

作者简介：周克江(1968－)，男，讲师、硕士，主研方向：移动计算，网格计算，网络安全
收稿日期：2006-05-24 **E-mail：**kejiang_zhou@hotmail.com

的主机后，主机可能捕获并篡改移动 Agent 的代码、数据和状态信息，将获取的机密数据泄落给其他 Agent，或者杀死 Agent，并终止服务等。针对这些安全性问题，移动 Agent 需要提供：(1)私有保护机制：加密保护敏感代码和数据，以防止信息泄露；(2)安全认证机制：建立可靠的移动 Agent 的通信实体(移动 Agent 与代理服务器，移动 Agent 与移动 Agent 之间的安全认证机制；(3)授权机制：通过授权以限制移动 Agent 对代理服务器的访问权限以及代理服务器对移动 Agent 的操作控制。

3 安全认证机制

下面介绍一个通用的移动Agent系统的认证模型。它支持基于历史的安全策略的定义和实施该策略使用SPL(Security Policy Language)[7]定义并被一个安全监控器执行在平台中可以定义一些基于历史的安全策略如Chinese Wall[8]这些策略在移动Agent案例中得到应用，因此Agent的操作可以基于Agent的过去行为被允许或被拒绝。

(1)安全策略的定义：安全策略的基础是 4 个模块：实体，组，规则和策略。规则是在实体/组之间通过关联建立的约束；策略产生于多重规则和组的组合。这种语言是面向策略和基于约束。实体代表具有明确接口的对象，这些对象可以被获取和修改，实体不仅可以表示为内部认证的模型化对象，也可以表示外部平台驻留的对象。尽管存在一些内部的实体如组规则或策略，但是多数是外部对象如移动 Agent 或文件。每个外部实体具有一个关联的类型，该类型用来定义它的接口和属性。对象和移动 Agent 的定义：

```
Type    Object
{
 String Name;
 User Owner;
 String HomeHost;
 Number timeOfcreation;
}
Type mobileAgent extend object
{
 Boolean running;
 String group previousHosts;
}
```

另一个重要的实体是规则。规则是建立认证操作约束的实体。一个认证策略可根据一组具有 3 个值的规则逻辑表达式来表示，它们有 3 个值：允许，拒绝和不使用。这些值决定了在系统模型中产生的事件的可接受性。实现基于历史的安全策略有两个重要的事件类：当前事件和过去事件。第 1 类事件是正被检验的和需要证明的事件；第 2 种类型的事件是已经被认可或已被拒绝的事件，这类事件用于构成对当前事件判断许可或拒绝的知识基础。一个规则由两个逻辑表达式组成。第 1 个是定义规则的适应域；第 2 个表达式是可接受性的域。一个规则的实例可描述如下：

```
DestinyRule:ce.source.type=mobileAgent &
            ce.source=”AgentJohn”&
            ce.target.host=”hostA”&
            ce.operation=”migration”::true
```

这个规则适用所有到主机 A 的移动请求，都由移动 Agent“AgentJohn”产生，可接受域总是真，事件也如此。策略是一个给定事件定义认证和禁止的一组规则和组。在一组完整的规则中，只有那些具有真正的可适应域才需要检查它们的可接受性域。

(2)认证体系。在系统模型中，应用程序建立并启动一个 Agent，它要通过一个 home 代理与其它 Agent 交互操作，这些操作既可以是 Agent 相应的方法，也可以是平台服务。系统平台支持大量的服务，如 Agent 的移动或复制。有一个移动代理与 Agent 的 home 代理协作，在 Agent 运行的主机定位，它可以管理 Agent 的运行流，在远程主机中移动代理扮演 home 代理功能。反之，home 代理 Agent 永久驻留在主机中，远程代理在主机之间是移动和传输。远程代理是由 home 代理在目标主机中创建的，代理的创建是基于 Agent 的代码和相关的安全策略、home 代理由系统模型的自动工具依次创建，程序员首先完成 Agent 的代码，接下来由用户(区别于程序员)定义一组安全策略，利用 SPL 创建一个关键的策略文件，根据这个策略文件和 Agent 代码，程序员可以执行模型的自动工具创建 home 代理，home 代理可以被任何应用程序使用。home 代理的结构如下图 1 所示。

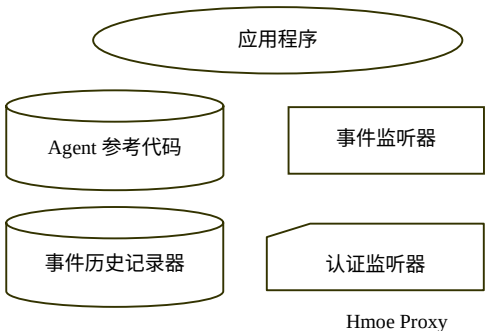

图 1 home 代理结构

home 代理的体系结构由 4 个模块组成：参考的 Agent 代码，事件监听器，事件历史记录器和认证监控器。Agent 参考是允许访问 Agent 完成 Agent 行为的代码和数据。当 Agent 移动到远程主机时这个参考被删除，因为 Agent 的代码和数据已经移动到远程主机。在新的 Agent 主机中由移动代理创建一个新的 Agent，这个代理接收该 Agent，启动一个用它的一个方法的执行流。Agent 的执行是从原始的主机移动到移动代理主机。事件监听器是一个事件操作的可靠模块，如移动、文件系统访问或 Agent 之间的请求等操作，该模块与认证监控器交互，通知它一个新的事件需要被认证，该模块也是移动的。因为它总是与 Agent 的代码和数据驻留在一起。事件监听器收集 4 类不同种类的事件：平台事件，操作系统(OS)事件，应用程序请求和 Agent 请求，平台事件是由系统模型产生并且是 Agent 操作的结果，如移动或复制。OS 事件是由访问系统资源产生的，如访问本地磁盘文件或网络端口。应用程序请求是调用 Agent 的可以控制它们的执行流的方法，如停止或恢复。Agent 的请求是一个 Agent 允许到另一个 Agent 的引用，是 Agent 的协作。认证监控器为了决定当前事件的可接受性检查一些属性，事件监听器可以建立一个事件的结构，并填入事件的所有属性。一旦给定 Agent 要验证它的可接受性，它将向前查找事件历史记录器检查它是否必须记录它，为了优化，有些事件不需要记录，事件历史记录器检查认证监控器(实施它的策略)是否不需要这个事件，如果回答是否定的，事件将被丢弃并不被记录，保持事件历史记录尽可能小。

（下转第 167 页）