基于TPM的移动代理安全密钥管理

武小平 赵 波 张焕国
(武汉大学计算机学院 武汉 430079)

摘 要 针对移动代理系统安全问题中的密钥管理,采用可信计算平台的核心硬件模块TPM提供的树形密钥管理体系,分析了其密钥的安全存储特性。在主机与代理之间的安全通信应用中,分析了安全会话密钥的生成与使用。

关键词 密钥管理,可信平台模块,移动代理安全

Secure Key Management of Mobile Agent Based on TPM

WU Xiao-ping ZHAO Bo ZHANG Huan-guo
(School of Computer, Wuhan University, Wuhan 430079, China)

Abstract Aiming at secure key management in mobile agent security, the key management tree architecture provided by core hardware module of trusted computing platform was adopted. The key's secure storage feature was discussed. On the application to secure communication between host and agent, generation and using of secure session key were also analyzed.

Keywords Key management, Trusted platform module, Mobile agent security

1 引言

移动代理技术的移动特性使得其适用于多种网络应用,如移动计算、智能网络,特别是电子商务等。但是,移动代理技术能否具有足够的安全性和稳健性是人们关心的首要问题。在移动代理系统中,代理的运行需要分布式系统中的远程主机为其提供执行环境。远程主机的所有者、代理所代表的用户以及移动代理软件的开发者都是不同的实体,这样就必然会存在安全隐患。

借鉴现代分布式系统中的许多技术和方法,根据移动代理系统的特点进行改进和扩充,现有的研究已取得了一定的成果,提出了许多提高移动代理系统安全性的方法[3,4,8]。综合这些解决方法来看,基于密码学方法的一些技术可以认为是其中的基本技术。例如为保护代理传输的安全,采用相应的数据加密、完整性校验和身份认证等技术可以减少网络传输中的攻击所带来的损失[6,7];移动代理出发时要求携带代理所有者的合法公钥证书和采用其私钥签字的一些信息(如时戳、有效期、用户需求等),远程主机再通过对代理所有者的公钥证书及签字的验证来认证代理的合法性,并根据移动代理的身份来提供相应的资源接入权限等。

从技术的角度看,移动代理现有安全方案中比较共同的关键问题之一就是密钥的安全管理。一旦密钥失效,则相应的安全策略也就没有什么意义了。密钥管理包括了密钥的产生、存储、分配、组织、使用、注销等一系列的技术问题,是一项严格的系统工程。传统的对称密码体制和非对称密码体制在进行密钥管理时,通常都会有一个可信的第三方的参与,其职责就是对参与的各方进行认证和为他们分派密钥。而这种集中式的管理又使得如何确保密钥管理中心和分配中心的安全可信和高效成为首要的问题。本文利用可信计算平台(Trusted Computing Platform, TCP)[1]的核心硬件模块(Trusted Platform Module, TPM)[2]提供的安全信任根和可信的存储根密钥(Root Of Trusted for Storage, RTS),结合TCP的分级密钥管理体系,实现移动代理系统中的安全密钥管理。

2 可信平台体系结构

2.1 硬件支持的信任根

可信平台模块TPM是可信计算技术的核心,是一个含有密码运算部件和存储部件的小型片上系统。作为可信平台的构件,TPM的组件能够正常工作。TCG给出了TPM组件的体系结构规范,本文利用其硬件支持来实现对上层移动代理系统的安全密钥管理。如图1所示,在TPM密码协处理器的支持下,基于硬件的随机数生成可以产生所需要的真随机数;密钥生成模块可以根据给定的相关参数(如密钥类型、密钥大小等)生成密钥对;平台配置寄存器PCR则用若干个160位的存储区来保存离散的完整性度量值;芯片内部的非易失性存储器则保存了TPM相关的一些永久性数据,并提供从外部的读写访问。

从概念上讲,TPM对其所在的平台创建3个可信的根,这些可信根用来建立平台的可信性和保证平台安全机制的实现。

到稿日期:2008-06-02 本文受国家863计划:可信PDA计算平台关键技术与原型系统研究(2006AA01Z442),武汉市青年科技晨光计划:可信计算平台上的移动代理安全研究(200850731373)资助。

武小平(1974—),男,博士,讲师,主要研究方向为可信计算、分布式安全,E-mail:wuxp@whu.edu.cn;赵 波(1972—),男,副教授,主要研究方向为信息安全;张焕国(1945—),教授,博士生导师,研究方向为信息安全。

可信度量根(Root of Trust for Measurement,RTM)可靠地度量任何用户定义的平台配置。RTM 作为可信的启动代码,是平台启动后最先执行的代码并且是不可改变的,因而是可信的。启动代码随着平台的启动,通过一种"信任传递"的过程将信任扩展到整个平台。在"信任传递"过程中,前一阶段的代码在把控制权传递给下一阶段前,先测量其度量值,并把度量值安全地保存起来,这个过程反复持续下去,直到可信的操作系统启动。可信报告根(Root of Trust for Reporting,RTR)允许经过验证的挑战者获取受 TPM 保护的区域中的数据,包括 PCRs 和非易失内存。并用签名密钥签名证实这些数据的真实性。其中 PCR 不仅保存数据,而且记录数据被保存的次序。可信的存储根(Root of Trust of Storage,RTS)保护所有委托给 TPM 的密钥和数据,基本上包括了密钥产生、密钥管理、加密和解密等功能。

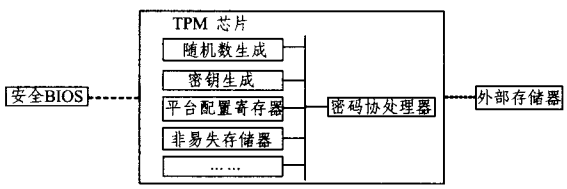

图 1　可信平台模块支持的密钥管理特性

2.2　可信软件接口

可信计算平台软件栈 TSS (TCG Software Stack)是支持平台可信模块的支撑软件,作为 TPM 芯片和用户应用之间的接口平台,支持 TPM 安全芯片向上提供平台认证、密码学服务和芯片管理等重要功能。TSS 内部涉及核心的对象和属性管理、上下文管理、授权与认证、安全操作、密钥管理和 TPM 管理等模块,是一套非常复杂的平台软件系统。

从可信平台的整个体系结构看,TSS 被设计成一个既不依赖于平台也不依赖于 OS 的软平台。其目标包括:基于 TPM 的功能为应用提供一个入口点;提供对 TPM 的同步访问;管理 TPM 的资源(包括创建和释放);给上层的应用程序提供一个合理有效的接口来替代 TPM 繁杂的指令流。

TSS 本身主要由两部分组成,分别是 TCS(TSS Core Service)和 TSP(TSS Service Provider)。TCS 驻留在用户态,通常以系统服务形式存在,它通过设备驱动接口(TCG Device Driver Library Interface,TDDLI)和 TPM 进行通信。TCS 提供了几乎所有的基本功能和复杂功能,像上下文管理、密钥管理、事件管理和审计管理等,它为 TSP 提供接口。TSP 通过 TCS 来使用 TPM 的功能。TSP 也提供了丰富的面向对象的应用接口,包括上下文管理、密钥管理和安全操作等。

TSS 支持本地过程调用(Local Procedure Call,LPC)和远程过程调用(Remote Procedure Call,RPC)两种程序调用方法。LPC 是直接来自另一个应用或模块的调用,这些应用或模块来自 OS 分配的同一进程或地址空间。RPC 通过在进程之间建立一种通信的基础设施并提供命名和地址的框架来实现进程间的交互。

2.3　密钥安全存储

前面提到的 TPM 的 3 个可信根之一的可信存储根 RTS,从其功能概念上看,其实就是存储根密钥(Storage Root Key,SRK)。SRK 作为存储的信任根被植入 TPM 内部。TPM 采用树形的密钥管理体系,如图 2 所示,TPM 只在其内部的易失存储器中管理并维护一小部分当前正在使用的密钥,对于那些没有被使用,或者成为不活跃的密钥则保存在 TPM 以外的存储设备中,只有需要使用的时候才通过 TPM 提供的接口函数(TPM_LoadKey( ))加载到 TPM。由于 SRK 的私钥部分被定义为不可泄漏于 TPM 之外,因此当需要 SRK 进行解密处理的时候,解密过程必须在 TPM 内部进行。这也就意味着如果一个被 SRK 加密的数据离开了所在平台,那么它就无法得到有效的解密。在图 2 的密钥管理树中,SRK 居于整个密钥树型结构的最上层,它通过对存储设备上密钥子树的根节点实施加密控制,并将这种能力向下委托,就能够实现对整个密钥体系进行有效的控制,外部存储的密钥也得到了较好的保护,从而整个密钥树的安全性是有保障的。值得注意的是,密钥的明文形式仅可能出现于 TPM 之内,同时密钥的使用需要一定的授权信息,因此密钥在整个体系中得到了非常好的保护,同时只有那些持有授权信息的使用者才能正确地使用密钥。

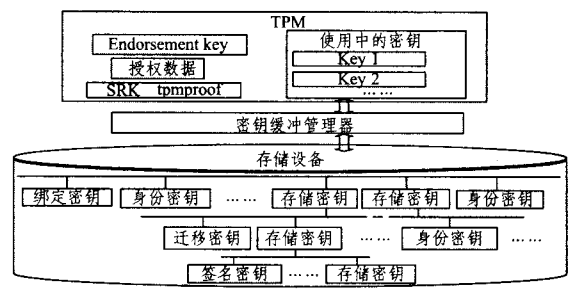

图 2　TPM 的密钥管理体系结构

TPM 通过提供 API 使得用户能够获得使用 TPM 相应功能的接口。一个 TPM 的新用户在使用 TPM 时,必须首先取得 TPM 的所有权,其中涉及设置所有者的口令、TPM 产生自己的存储根密钥 SRK 和主秘密 tpmProof 等,也即初始化的过程。为了开始使用 TPM,用户必须产生 TPM 身份密钥和一些加密密钥。密钥管理模块提供包括:产生不同类型密钥、赋予和修改密钥口令及迁移密钥到其他实体的 API。不同类型的密钥包括:存储密钥、签名密钥、身份密钥和绑定密钥等。

TPM 提供的 RTS 类 API 进一步可分为:①保护存储类命令。这类命令主要涉及到在 TPM 端点用公钥加密算法来为私钥操作准备相关数据和密钥,并且执行这些私钥操作。②密钥管理类命令。负责密钥的产生和密钥的使用,它是 TPM RTS 的重要组成部分。每种密钥都有严格的使用范围,不同类型的密钥常常与不同的 API 绑定,尽量减少密钥类型对 API 的混合使用。所有被 RTS 管理的密钥分为可迁移和非迁移两类型。非迁移密钥只能和一个 TPM 绑定,典型例子为身份证明密钥(Attestation Identity Key,AIK),如果 AIK 可以迁移,则可以允许一个 TPM 冒充另一个 TPM。签注密钥(Endorsement Key,EK)、AIK 和 SRK 对 TPM 是不透明的,且 SRK 和 EK 是嵌入到 TPM 的。③密钥迁移类命令。迁移模块的作用是把可迁移密钥迁移到其它 TPM。

TPM 的密钥都有使用口令和迁移口令。使用口令授权使用密钥进行加密解密操作,但不能恢复出密钥的私钥。只有在 TPM 内部才恢复出私钥的明文,一旦密钥离开 TPM 均以密文的形式存在。

2.4　TSS 密钥管理

在 TPM 的密钥安全存储体系基础上，TSS 的密钥管理服务也定义了相应的分级永久性密钥管理树形结构来支持对上层的应用，如图 3 所示。密钥管理服务接口的设计支持灵活的密钥结构，比如，对一个部门来讲，允许定义级别较高的安全密钥，也可以包含级别较低的或临时可迁移的密钥等。所有被 TSS 密钥管理服务中心管理的密钥必须在 TCS 或 TSP 的永久存储数据库中注册。每个注册的密钥都将分配唯一的标识 UUID，这些 UUID 只是一个标识，通过它调用者并不能获得密钥注册的系统的任何信息。系统仅可以通过 UUID 在永久存储区中查找它的公钥信息和它的父密钥信息等。子密钥的存储依赖于父密钥，因为它的私钥部分是由父密钥加密后存储的。依此向上递归，就追溯到存储根密钥 SRK，存储根密钥用于为 TPM 保证密钥和数据是可信的。

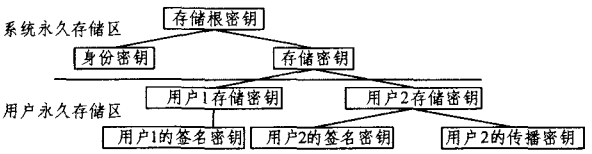

图 3　TSS 密钥树结构

以创建用户 User1 的签名密钥为例，有如下几个步骤：

1）当拥有者得到装有 TPM 的平台后，可通过初始配置得到 TPM 的 owner 权限，生成 SRK，也就是得到存储根密钥，SRK 存储在 TPM 内部，它不以明文的形式存储在外部。

2）然后通过一系列的操作，得到 TPM 的拥有权限，进行创建存储密钥、身份密钥。根据不同平台的类型不同，可以分为一般的固定平台存储密钥和移动平台密钥。它由系统创建，用于对用户密钥进行加密操作。

3）当不同的用户使用 TPM 时，需要由系统为其分配一个用户的存储密钥，并对其进行授权配置。

4）当用户需要使用密钥进行签名时，则向 TPM 发送请求，要求创建用户签名密钥，TPM 以此用户的存储密钥作为父密钥创建一个新的子密钥即用户的签名密钥。

3　安全分析

主机与代理之间的认证

以基于非对称密码体系的典型认证过程为例。在可信平台上加载的代理平台通过 TSS 密钥管理服务系统永久存储数据库中注册有自己唯一的用户 ID 和密码，并拥有自己的密钥/公钥对，每个密钥都分配有唯一的标识 UUID。代理在移动至可信平台上的代理平台之前，先由系统通过 UUID 在永久存储区中查找代理平台的公钥信息和它的父密钥信息等，并告知自己的公钥。由前面的描述可知，基于 TPM 的密钥树是可信的。双方用彼此的公钥加密，对方用自己的私钥验证，经过两次握手，双方达到认证信任。

主机与代理之间的安全通信

为建立移动代理与代理平台之间的进一步通信，双方需要在不可信的网络上建立一个安全信道，于是需要生成会话密钥来用于对消息进行加解密和哈希运算[5]。该会话密钥在可信平台上由以 SRK 为根的密钥树为基础来生成以确保其安全可信并由双方共享。平台上的其他实体未经授权是无法获知该会话密钥的，从而保障了通过该安全信道传输消息的互信。

主机与代理之间安全通信的建立及使用的挑战应答协议如图 4 所示。

A（Mobile Agent），P（Platform）
$A \rightarrow P$: ID_A；请求通信初始化 $P \rightarrow A$: N_P；128bit随机数 $A \rightarrow P$: N_A；128bit随机数
生成通信密钥 P，A: $K_{P \rightarrow A}$ =MAC_{SRK} [C_{comm}，C_P，N_P，N_A] A，P: $K_{A \rightarrow P}$ =MAC_{SRK} [C_{comm}，C_A，N_A，N_P]
发送应答 $P \rightarrow A$: $Response_P$=MAC $K_{P \rightarrow A}$ [N_P，N_A] $A \rightarrow P$: $Response_A$=MAC $K_{A \rightarrow P}$ [N_A，N_P] 应答校验 P: MAC $K_{A \rightarrow P}$ [N_A，N_P]=? $Response_A$ A: MAC $K_{A \rightarrow P}$ [N_P，N_A]=? $Response_P$

图 4　代理与平台之间会话密钥生成的挑战应答协议

出于安全的考虑，移动代理和平台双方都可以发起协议初始化，双方各自选择一个随机数，设定一初始化常数和各自的密码常数，结合可信平台上的以 SRK 为根的密钥树，生成会话密钥 $K_{A \rightarrow P}$ 和 $K_{P \rightarrow A}$。然后双方用生成的密钥互发消息认证码 MAC。通过对 MAC 的校验确认通信期间没有重放攻击和中间的修改，并由双方共享密钥信息。从图中会话密钥生成部分可以看出，这种校验也使得双方都生成单向的密钥，一个用来加密发送消息，另一个用来解密和检查收到的消息。并且这种会话密钥也可以用于以标准的 TSL 协议进行的安全通信。

结束语　在基于移动代理的分布式应用系统的安全机制采用的若干技术中，有主机与代理之间的认证、移动代理的访问控制、移动代理的安全传输以及移动代理的安全路由等，技术基础就是基于密码学的相关技术，而密钥管理又是其中的前提和基础。无论是采用传统的对称密码体制和非对称密码体制，一旦密钥不能保证安全，则安全基础不复存在。各种不同的密钥管理方式，其安全强度也各不相同，在条件许可的情况下，安全强度最高的硬件平台方式是一个较好的选择。任何可信都是建立在某一个层次上的，如果你能直接访问更低的层次，那么上一个层次的保护将是毫无作用的。传统的系统中，密钥和授权信息都直接存储在内存和硬盘之中，攻击者有很多的方法来获取它们。在可信平台的体系中，所有这些秘密数据都是由 TPM 来保护的。于是，只有攻击者能够攻破 TPM 才能攻破系统的防护。这样，TPM 成为了系统可信的最低层次，它提供了整个系统可信的基础。可信计算平台以 TPM 为硬件核心模块，定义了各种不同类型的密钥并严格限制了其属性和使用方法，包括加解密方案、签名方案、密钥长度和算法类型等。提供以 SRK 为根信任的树形密钥管理体系，为移动代理的密钥管理提供一个可信的第三方，并用防篡改的硬件进行存储保护，都极大地加强了密钥管理的安全强度。

参 考 文 献

[1] Trusted Computing Group. TCG Specification Architecture Overview, Specification Revision 1.4[OL]. http://www.trustedcomputinggroup.org. 2007.8

[2] Trusted Computing Group. TCG Software Stack (TSS) Specification Golden Candidate 2 Errata 3c. November 2005; TSS Specification Version 1.2 Level 1 Errata A, Part 1: Commands and Structures, March 2007

（下转第 78 页）

$\frac{\omega(1-\beta+\beta^3)}{(c-a)(1-\beta^2)^2}$，由式(11)得网络 j 获得最大效用时所提供的带宽 $b_j^* = \frac{\omega(1-\beta-\beta^2)}{(c-a)(1-\beta^2)^2}$。

3 仿真结果及分析

仿真中，参数的设置如下：$a=4.5$，线性价格惩罚因子 $k=1/14$，非线性价格惩罚因子 $\omega=5$，网络之间相互影响的系数 $\beta=0.1$，线性价格方案下单位带宽的成本支出 $c=2.5$，非线性价格方案下单位带宽的成本支出 $c=5$。

图 1 描述了带宽之间以及带宽与效用的关系。图中左上部分描述的是网络 i 与网络 j 所提供带宽之间的关系，从图中可以看出网络向用户提供的带宽量是一个“此消彼长”的现象，这就避免了网络之间由于盲目竞争而造成带宽资源的浪费。

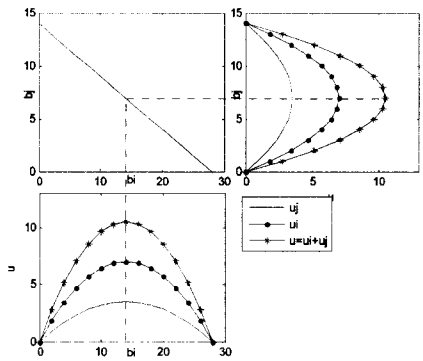

图 1 带宽之间以及带宽与效用的关系图

左下部分图描述的是网络 i 所提供的带宽与效用函数之间的关系，由图可知网络 i 所提供的带宽 $b_i^* = 14$ 的时候，保证了各网络以及网络整体取得最大效用。观察到网络效益随着带宽的增加而增大，当效益达到某一最大值后就会衰减，这是因为当所提供带宽不断提高后，网络对带宽的定价就会随之降低，以吸引更多的“购买者”，从而使网络获得的收益降低。基于这方面考虑，Stackelberg 博弈避免了网络无限度地增加所提供的带宽量，从而提高了网络的利用率。同理，右上部分图描述的是网络 j 所提供的带宽与效用函数之间的关系。

另外，从图 1 可知网络 i 保证各网络在 $b_i^* = 14$ 时取得最大效用，对应于网络 j 的带宽 $b_j^* = 7$。根据 Stackelberg 博弈的主从关系可知，网络在 $b_j^* = 7$ 时同样取得最大效用，图中仿真结果证明了理论推导的正确性。这是因为网络 j 是在观测到网络 i 所选策略（保证效用最大时的带宽量）后选择提供 b_j 的带宽，因此它一定能保证在对应时刻网络效用取得最大。

由图 2 得知，当采用非线性价格策略时，价格在带宽比较小的时候就能很快降下来，从而刺激用户更多地使用带宽。当带宽增加到一定程度的时候，价格的下降速度就明显地慢了下来，这就避免了用户因为过于廉价而无限制地购买带宽造成网络资源的浪费，从而达到了合理利用网络资源的目的。

图 3 对比了在分别采用线性和非线性价格策略时，网络所获得的效用。由图可知，当采用线性价格策略时，在达到最大效用以前效用曲线比较平滑，激励效果比较小。相比之下，采用非线性价格策略，网络效用能够在达到最大值之前以指数形式陡增，激励作用明显。因此，在网络带宽供大于求的时候适合采取非线性价格策略，激励用户更多地购买带宽。

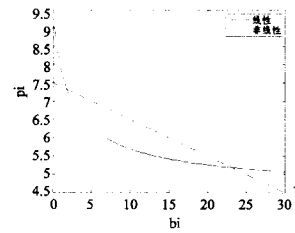

图 2 线性/非线性价格曲线

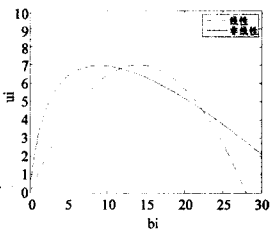

图 3 线性/非线性效用曲线

结束语 负载均衡对于提高网络性能，保证业务 QoS 是非常重要的。本文利用 Stackelberg 博弈模型解决了多主接入(Multi-Homing)中的负载均衡问题，从线性和非线性价格方案两个角度，实现了网络效用最大化，扩大了算法的适用范围。仿真结果表明，完美子博弈均衡解能够保证网络获得最大收益值，实现了带宽资源的合理分配。本文中为了使仿真结果更加具有直观性，价格惩罚因子的选取具有特殊性。如何选取合适的价格惩罚因子，将是实际应用中需要做进一步研究的工作。

参 考 文 献

[1] Lu Kai, Zomaya A Y. A hybrid policy for job scheduling and load balancing in heterogeneous computational grids[C]// IEEE Parallel and Distributed computing, 2007. ISPDC '07. sixth international Symposium. July 2007

[2] Dhakal S, Hayat M M, Jorge E, et al. Dynamic load balancing in distributed systems in the presence of delays: a regeneration-theory approach[J]. IEEE Transactions on Parallel and Distributed Systems, 2007, 18(4): 485-497

[3] Lee Sang Hoon, Han Youngnam. A novel Inter - FA handover scheme for load balancing in IEEE 802.16e system [C]// Vehicular Technology Conference, 2007. VTC2007-Spring. IEEE 65th. April 2007: 763-767

[4] Niyato D, Hossain E. Integration of WiMAX and WiFi: optimal pricing for bandwidth sharing [J]. IEEE Communication Magazine, 2007, 52(7): 140-146

[5] Fudenberg D, Tirole J. Game Theory[M]. 北京：中国人民大学出版，2003: 59-120

（上接第 67 页）

[3] Wu Xiaoping, Shen Zhidong, Zhang Huanguo. The Mobile Agent Security Enhanced by Trusted Computing Technology[C]// The 2nd International Conference on Wireless Communications Networking and Mobile Computing. Wuhan, IEEE Press, 2006

[4] Borselius N. Mobile agent security[J]. Electronics & Communication Engineering Journal, 2002(10)

[5] Lee R, Kwan P, McGregor J P, et al. Architecture for Protecting Critical Secrets in Microprocessors[C]// Proceedings of the 32nd International Symposium on Computer Architecture (ISCA 2005). June 2005

[6] Lee H, Alves-Foss J, Harrison S. The Use of Encrypted Functions for Mobile Agent Security[C]// Proceedings of the 37th Hawaii International Conference on System Sciences. IEEE Press, 2004

[7] Zhong S, Yang Y R. Verifiable Distributed Oblivious Transfer and Mobile Agent Security, Mobile Networks and Applications [J]. Springer Science, 2006(11): 201-210

[8] Foster D, Varadharajan V. Security and Trust Enhanced Mobile Agent based System Design[C]// Proceedings of the Third International Conference on Information Technology and Applications (ICITA'05). IEEE Press, 2005