签证机关管理系统*

佘 坤 周明天 蔡 兵

(电子科技大学微机所,成都 610054)

摘要 虽然公钥体制中的公开密钥常存放于公开目录中,但并不是说公开密钥不需要保护.本文介绍了公开密钥所面临的威胁及建立公开密钥管理体制的必要性,并分析和设计了这样一种体制——签证机关管理系统,且在UNIX环境上实现了其简单而实用的模型.

关键词 智能卡,数字签名,签证机关,证书.

近几年计算机的迅速普及和通信技术的发展,使人们利用计算机的大量存储资源和网络通信能力进行信息交流不再成为问题.资源的共享也增加了"恶意"的入侵者窃取、篡改或冒充的机会.因此,必须在系统中增加适当的安全机制防范这些安全攻击,如存取控制、数据加密、鉴别、数字签名和数据完整性等等,而密码技术是常用的安全技术之一.

无论是使用常规密码体制还是使用公钥体制的安全系统,秘密密钥在产生、分配、存储和传输过程中都面临泄漏的威胁,因此保护和管理秘密密钥是十分重要的课题.然而,这并不意味着公钥体制中的公开密钥不需要保护.用公钥体制实现鉴别、抗抵赖等安全服务时,修改公开密钥可能使系统遭致冒充、抵赖等攻击,保护公开密钥也同样重要.

本文在UNIX系统上,设计和实现了签证机关管理系统的一个简单模型,管理和维护公钥体制的秘密密钥和公开密钥.

1 公钥体制和数字签名

1976年,Diffie和Hellman引入了公开密钥密码体制的思想[1].由于公钥体制特有的性质,不仅使公开信道上机密信息的传递易于实现,也方便了数字签名的实现,使系统能在开销不大的前提下完成鉴别、数据完整性和抗抵赖等安全服务,安全的开放系统成为可能[2].

公钥体制中,每个用户拥有两种密钥:一个公开,一般存放在公共目录中,任何人都可访问,称为公开密钥;另一个保密,只有用户本人知道,称为秘密密钥,常用于标识用户身份.

假设有一用户A,我们用符号Ap,As分别表示A的公开密钥和秘密密钥.由于它们与加密、解密变换的一一对应关系,我们也用它们分别表示加密、解密变换,则消息明文m的

* 本文1993-08-16收到,1993-12-13定稿

本文得到国家电子科学基金项目的支持.作者**佘坤**,1967年生,助教,主要研究领域为信息安全,开放系统计算.**周明天**,1939年生,教授,主要研究领域为开放系统环境,分布计算系统与软件.**蔡兵**,1969年生,助教,主要研究领域为容错计算.

本文通讯联系人,佘坤,成都 610054,电子科技大学微机所

加密表示为 Ap(m)=c (c 是相应密文);对 c 的解密表示为 As(c)=m.

当选用的密码体制具有对称性,即满足 As·Ap=Ap·As 时,我们用 As 加密明文 m:As(m)=c,它具有如下特性:(1)任何人都可用 Ap 对 c 解密得到明文 m:Ap(c)=m,从而确定 c 来自用户 A;(2)除 A 外,其他任何人不能伪造 c;(3)A 不能否认曾用 As 对 m 加密.

特性(1)使公钥体制适用于消息源鉴别、身份鉴别等;特性(2)使之可用于数据完整性服务;特性(3)使之能应用于抗抵赖服务.

实际使用中,我们常常利用数字签名技术实现这些服务.例如,A 将 A{m}=(m,As[h(m)]),发给用户 B,其中 m 表示消息 m;A{m}表示 A 对 m 数字签名;h(m)表示将单向 Hash 作用于消息 m;As[h(m)]表示 A 用其秘密密钥对 h(m)加密.B 首先从目录中获得 Ap,计算 Ap(As[h(m)])=h′(m),然后取 m 计算 h(m),再检测:

if h(m)=h′(m) then　确定 m 来自 A 且没有被篡改;

else　签名可能被修改或伪造.

2　公开密钥的安全威胁

数据的安全威胁主要有 2 个方面:泄漏和篡改[3].对秘密密钥来说,主要考虑如何防止泄漏.因为秘密密钥的篡改,会使世界上任何人(包括该密钥的主人)都无法解密公开密钥加密的信息,不会威胁系统的安全.对公开密钥来说,我们一般将它存放在公共目录,该目录任何人都可以访问,任何人都可以修改.假设实体 A 对一段消息 m 数字签名:

$$A\{m\}=(m,As[h(m)]),$$

并将它发给实体 B.m 中应包括 A 的标识号 ID_A,否则,B 不知道谁对 m 签名.令

$$m=(ID_A,M),(M\text{ 是要鉴别的原始明文})$$

则

$$A\{m\}=A\{(ID_A,M)\}=((ID_A,M),As[h(m)]).$$

如果在 B 验证前,实体 C 修改了公共目录中 A 的公开密钥,则 B 确定 m 被修改;如果 C 用自己的公开密钥 Cp 代替目录中 A 的公开密钥 Ap,并对消息 m 重新数字签名:

$$C\{m\}=((ID_A,M),Cs[h(m)]).$$

然后发给 B,由于 m 中身份标识是 A 的标识号,B 从目录中取 Ap(其值为 Cp)验证签名,将确定消息来自 A 且没有被修改,则 C 冒充 A 成功.事后,C 又将目录中 A 的公开密钥改回原值,不会留下一点迹象.

同样,如果 B 使用 A 的公开密钥 Ap 加密,而 C 已经将 Ap 改为 Cp,则 C 将获得 A、B 之间的机密信息[4].因此,公开密钥的可靠性也是安全系统必须考虑的问题.

3　签证机关管理系统

3.1　签证机关(CA)

因为数字签名也可应用于数据完整性服务,本文直接使用它保护公开密钥.例如,A 的公开密钥由实体 X 签名:X{Ap}=(Ap,Xs[(Ap)]),并代替 Ap 存放在目录中.任何人只要使用 Xp 验证 X{Ap}即可知道 Ap 是否被篡改.因此公开密钥的签名方(这里是 X)必须是可信的、没有恶意的实体[2,5,6].我们称此可信的实体为签证机关[5],记为 CA.同样,CA 的秘

密密钥和公开密钥也需要保护,为此我们应该设计一个专门的系统,对各实体(包括 CA)秘密密钥和公开密钥的产生、分配、存储等进行控制和管理,我们称该系统为签证机关管理系统,记为 CAMS.

CA 应该是隔离的、物理安全的、"脱机"的实体[5],象一个密码装置,CA 的秘密密钥存放于该装置中,只有 CA 自己知道,不可能泄漏. 由于智能卡(smartcard)独有的特性:物理安全和逻辑安全[7],我们建议使用智能卡做 CA,数字签名算法直接集成在卡中,秘密密钥存放在卡内保密区. 另外,假设每个用户也拥有同样功能的智能卡.

3.2 证　书

CA 对公开密钥数字签名的结果称为证书. 实际上,证书不仅包含公开密钥,还有其它用于验证和防止重放的信息. 证书表示为 $CA\ll A\gg=CA\{senum,sgndata,CA,A,Ap,T^A\}$,其中 senum 表示证书的编号,在同一 CAs 签名的证书中应保持唯一;sgndata 表示 CA 生成签名的算法标识符;CA,A 分别表示 CA 和 A 的标识;Ap 表示 A 的公开密钥;T^A 表示证书的有效期.

由于 CA 是脱机的,用户必须与 CA 事先约好(如通过电话或电子邮件等),才能请求 CA 签名证书. 一般我们假定用户是了解其 CA 的公开密钥的,具体过程如图 1.

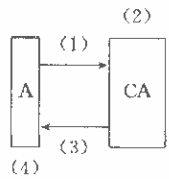

图1 证书生成过程

(1) 用户 A 将 $A\ll A\gg$ 发给 CA,申请签名;

(2) CA 从 $A\ll A\gg$ 中取出 Ap,验证 $A\ll A\gg$ 的可靠性;如果成功,CA 修改编号和有效期,用 CAs 对 Ap 重新签名,生成 $CA\ll A\gg$,转(3);否则,CA 通知 A 重发 $A\ll A\gg$,过程返回(1).

(3) CA 将 $CA\ll A\gg$ 存入目录 A 项中,并返回给 A 一份拷贝;

(4) A 接收 $CA\ll A\gg$ 后,用 CAp 验证证书可靠性. 如果成功,将它存在本地存储介质中(如智能卡);否则,请求 CA 重新发送.

现在,任何其他用户都可以访问目录获得用户 A 的公开密钥,但不能修改.

3.3 二级签证树模型

CA 的公开密钥与用户的公开密钥一样,是公开的,存放于公共目录,也需要保护. CA 的证书应由 CA 的 CA 发行. 从 CA_0 到 CA_n 的证书序列:

$$CA_0\ll CA_1\gg,CA_1\ll CA_2\gg,\cdots,CA_{n-1}\ll CA_n\gg$$

构成了 CA_0 到 CA_n 的签证路径,这些 CA 对 CA 产生的证书称为交叉证书. 任何了解 CA_0 公开密钥的人都可以依次验证签证路径中每一证书而验证 CA_n 公开密钥的可靠性. 假设 A 是 CA_n 的一个用户,它的证书是 $CA_n\ll A\gg$,该证书与 CA_0 的签证路径构成了 A 的证书链. 同样,任何了解 CA_0 公开密钥的实体都可依次验证证书链中每一证书而验证 Ap 的可靠性:

$$A_p=CA_{0p}\cdot CA_0\ll CA_1\gg,CA_1\ll CA_2\gg,\ldots,CA_{n-1}\ll CA_n\gg,CA_n\ll A\gg.$$

一般 CA_0 是验证的公共可信点,构造签证路径(或证书链)的关键在于找到此公共可信点. 不同的验证方,可能有不同的公共可信点,需要构造不同的签证路径. 为了构造签证路径,用户必须详细了解 CA 的拓扑结构,构造签证路径的难度与 CA 的拓扑结构有关. 为了简单,我们将各 CA 按树型层次组织起来,最高一级 CA 称为根 CA,记为 Root—CA,并称此树为签证树. CCITT 讨论了签证树中构造签证路径的几种情况[5],为了简化签证路径,降

低验证开销，使用了交叉证书、向后证书等技术，但如何从众多的证书中选择最佳签证路径，仍然很困难，证书越多，难度越大.

J. K. Omura 引入了一个签证中心的思想[6]，任何用户或 CA 都知道签证中心及其公开密钥，这只要查一下目录或电话簿等，甚至存放在每一智能卡中. 其实，该方法相当于一级签证树模型，签证中心是所有用户的 CA. 优点是不需要签证路径，证书链只包含一个用户证书，验证过程简单；缺点是它可能会成为网络的一个"瓶颈"，只能用于用户不多的情形下.

鉴于此，本文建议选签证树 Root－CA 作为所有验证的公共可信点，任何用户都知道 Root－CA 及其公开密钥. 所有签证路径都可以从 Root－CA 出发来构造，与验证方无关. 该算法简单，不要求使用向后证书. 我们称此路径为向前签证路径，记为 FCPath.

显然，每一 CA 的所有用户都有同一 FCPath，用户不必再详细了解签证树，可以由 CA 统一完成 FCPath 的构造，然后交给各用户构造各自的证书链.

证书的可靠性与签证树的规模、层次无关，仅仅由 CA 的可信度来保证. 所以，实际使用时，我们把签证树层次只定为二级，所有 CA（除 Root－CA 外）都是 Root－CA 的子女（如图 2 所示）. 这样做不仅不会破坏证书的可靠性，反而会大大减少构造 FCPath 的复杂度及其验证开销.

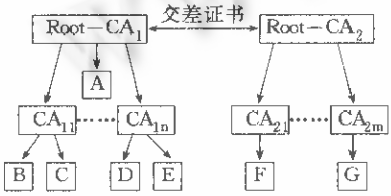

图2 二级签证树模型

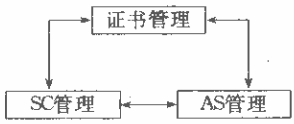

图3 CAMS管理模块

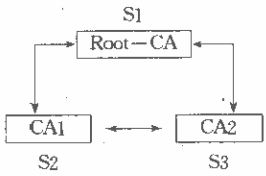

图4 二级CAMS的系统配置

例如，在图 2 中，用户 B、C 和 CA_{11} 的 FCPath 只包含一个证书 Root－CA_1≪CA_{11}≫，即每个 CA 获得其公开密钥的证书（Root－CA 签名），立即就可构造 FCPath，并发给该 CA 的每一用户. 由于 FCPath 的构造及其验证开销小，同一签证树各 CA（不包括 Root－CA）之间不再需要交叉证书，它一般只用于不同签证树的 Root－CA 之间，以使不同树的用户也能安全通信.

如果 D 要确认 Cp 的可靠性，首先他必须获得 C 的证书链 Root－CA_1≪CA_{11}≫、CA_{11}≪C≫，这可以由 C 发给 D，则 D 从目录中获得 Root－CAp，计算看是否 Cp＝Root－CA_{1p}·Root－CA_1≪CA_{11}≫，CA_{11}≪C≫即可.

3.4 模块化设计

整个 CAMS 根据需要可分为智能卡（SC）管理，证书管理和鉴别服务器（AS）管理三大模块，它们之间是相互依赖的关系（如图 3）.

在我们的系统中，使用了 3 台 SunSparc－2 工作站（图 4）.

我们在 S1 上实现了 Root－CA，在 S2、S3 上分别实现了 CA_1 和 CA_2；各机上用户选本地机上 CA 作为自己的 CA.

·SC 管理模块　由于目前 SC 技术还不太成熟，我们对此进行了软件仿真. 这样，各个 CA 及工作站上所有用户的 SC 都是软件实现的.

SC 仿真程序必须完成 SC 的功能：包括① 密钥的本地生成、维护和取消（包括 DES 密

钥和 RSA 密钥)；② 建立本地 SC；③ 存储证书、FCPath 等可读数据；④ 集成 DES、RSA 加密算法和 RSA－sqMODn 算法.

• 证书管理模块　证书管理模块完成：① 证书的生成、验证；② 证书登记到 AS；③ CA 建立 FCPath，并传递给它所有用户；④ Root－CAp 由 Root－CA 自己签名，并登记到 AS 及传递给各 CA 和用户.其间所有通信(如用户与 CA 之间的证书传递)由 Sunlink(X.25)提供的 socket 通信机制完成.

• AS 管理模块　AS 在这里主要起数据库作用，存放公开密钥和证书等.我们使用 Sun 提供的类似 X.500 功能的 NIS 服务.NIS 中每一项使用全局用户名作关键字查询，属性有 X.25 地址、证书等.

4　小　结

签证机关管理系统使用数字签名和智能卡技术，涉及公钥体制密钥对、FCPath 和证书的管理，是鉴别交换、数据完整性和抗抵赖服务的基础.本文所述的二级签证树模型简单实用，速度快，所有模块使用了约 10000 多行 C 语言语句，本系统经运行，效果良好.目前正在此基础上开发强鉴别服务和安全的 MHS.

参考文献

1　Diffie W, Hellman M E. New directions in cryptography. IEEE Trans. on Info. Theory, 1976, **IT－22**: 644－654.

2　Diffie. The adolescence of public－key cryptography. Advances in Cryptology－EUROCRYPT'89, 1989. 2－6.

3　Denning D. Cryptography and data security. Addison－Wesley Publishing Company, Inc., 1982.

4　Denning D. Protecting public keys and signature keys. IEEE Comp., Feb. 1983. 27－35.

5　CCITT X.509. Directory－authentication framework. 1988.

6　Omura J K. Novel application of cryptography in digital communications. IEEE Communications Magazine, May 1990, **28**: 21－29.

7　Ferreira R C. The smart card: a high security tool in EDP. Philips Telecommunication and Data Systems Review, 1989, **47**: 1－19.

CERTIFICATION AUTHORITY MANAGEMENT SYSTEM

She Kun　Zhou Mingtian　Cai Bing

(*Microcomputer Research Institute, University of Electronic Science and Technology, Chengdu* 610054)

Abstract　Public－keys in a public－key system are usually registered in a public directory. This paper describes the threats faced by the public－keys and necessity of establishing public－key management system. It also analyses and designs one instance——certification authority management system, whose simple and practical model has been implemented on UNIX machines.

Key word　Smart card, digital signature, certificate, certification authority.