一个安全的电子邮件子系统

王 剑 波

(北京交通管理干部学院 101601)

电子邮件将成为人们的主要通信手段之一. 正如 Internet 网的管理人员指出的那样,国际大网 Internet 自开通后最为活跃的服务项目一直都是电子邮件. 由于电子邮件日益成为人们生活中的重要部分,它的安全性也就成了一个紧迫的话题. 人们寄信时总要把信纸封在信封里,因为他们不希望别人看到信的内容. 同样,人们在发电子邮件时也不希望第三者能够读到它. 对于要害部门或处于激烈竞争中的企事业来说,更是把电子邮件的保密性放到了至关重要的地位. 在这样一种背景下,我们设计了一个安全的电子邮件子系统 Sem(Secure subsystem of e-mail).

Sem 作为一个模块化的子系统,可以挂接在任一特定的电子邮件系统上,尽管设计时是以 Internet 上的 e-mail 系统为实验背景的. Sem 具有一个运行在 MS-Windows 下的通信接口及用户界面. 通信接口的主要功能是提供 modem 通信机制,大部分的 e-mail 系统可以通过 modem 以拨号形式进行邮件的接收和发送.

Sem 的用户界面以直观的方式提供了加密、解密、签名、鉴别、接收、发送、密钥管理等操作.

前面已提到 Sem 的安全机制是基于公钥体系的. 更确切地说,Sem 是 RSA 公钥体系与传统单密钥加密方案的混合体. 采用混合方案的主要原因是纯粹 RSA 体系的效率太低. 软件实现的 RSA,其加密解密的速度要比传统算法(如 DES)慢二到三个数量级. 在混合方案下,当用户 A 欲将某一电子邮件发送给用户 B 时,他首先用单密钥 K 按传统加密方案将邮件加密;然后从系统的公钥服务器取得 B 的公钥 KeB,用此公钥加密临时的单密钥 K,并将加密后的密钥 K 连同加密了的电子邮件一起发送给 B. 除此之外,A 也可以给电子邮件附上他的签名,以便接收者 B 可以确信他收到的是由 A 发出的邮件.

上面提到的公钥服务器是系统内管理所有用户公开密钥的一个管理中心. 它的作用包括公钥的收集,发布,更新及防止被窜改. 公钥可能被篡改,是公钥体系的一个较大弱点. Sem 在这一方面给出了自己的保护措施.

下图给出了 Sem 的总体结构. 后面各节将对图中的各个主要部分进行具体描述.

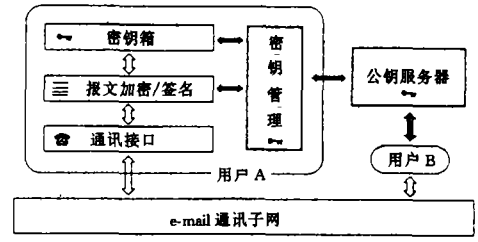


图 1 e-mail 安全子系统(Sem)

通信接口

Sem 的通信接口提供了在 Windows 下的终端仿真和 modem 通信功能. 由于相当一部分的 e-mail 系统是由运行于 UNIX 的 mail 例程演化而来的,因此 Sem 允许用户以拨号终端的方式访问 UNIX 主机并使用主机上的 mail 或其他邮件处理程序. Sem 仿真的终端包括 ANSI 终端,VT-1000,VT-52 等.

对于拨号接入的电子邮件系统,Sem 提供了内置的 modem 通讯功能. Sem 可以设置通信方式;通过 modem 自动拨号;提供自动发送和接收功能. Sem 也能对接收到的邮件进行识别,以判定是否属于加密文本. 对于加密邮件,还将自动检验其完整性,以避免由于传输错误而带来的影响.

通常的 e-mail 系统只能接收和发送 ASCLL 文本. 但是加密后的文本则是以二进

制原始字节的形式存在的. Sem 采用了一种类似于 Internet PEM (Privacy-Eahanced Mail) 的格式,用以把加密后的二进制字符转换成通常 e-mail 系统可接受的可打印 ASCLL 字节. 这种转换的核心思想,是用四个一组的 ASCLL 可打印字符来代替三个一组的二进制字符.

加密与签名方案

1. RSA 算法简述

由 Rivest, shamir, Adelman 发明的 RSA 算法是一个基于"大数分解"难度问题的公开密钥系统. 下面简述 Sem 所采用的 RSA 体系.

① 选择至少 512bit 长度的两个质数 p, q, 令 $n=p\times q$;

② 令 Ke 为加密密钥,Kd 为解密密钥,M 为明文,C 为密文,则有:

$$C=M\ \ Ke\ mod\ n,\ M=D\ \ Kd\ mod\ n,$$

实际上密钥对 Ke,Kd 中的任一个都可解开由另一个加密的密文. Kd 和 Ke 满足

$$M\ \ KdKe=M\ \ mod\ n$$

为提高 RSA 算法的效率,Sem 采纳了由 Quisquater 等提出的基于"中国剩余定理"的解密算法.

用户将整数对 (Ke,n) 公开. 任何欲给此用户发邮件的人都可以利用他的公开密钥加密. 公开密钥可以从公钥服务器得到.

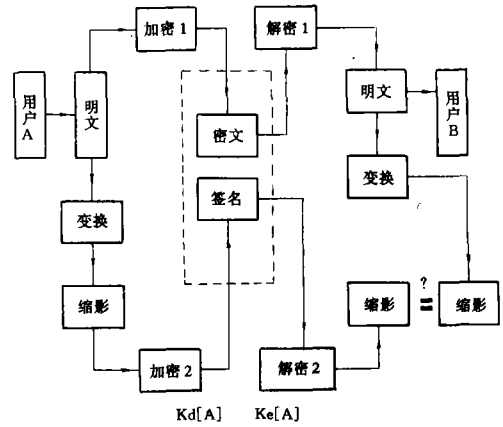
2. 单密钥加密算法

曾被广泛使用但正面临死亡的著名加密算法 DES,就属于单密钥加密算法. 这种算法只有一个密钥 K. 即加密和解密都要用到这个密钥. 显然,通信双方事先必须通过可信信道传递这个密钥,然后才能用它进行邮件加密.

在 Sem 中,为提高邮件加密速度,采用了一种具有较强安全性的分组加密算法. 并用 RSA 算法对分组加密中使用的单密钥进行加密,然后连同加密后的邮件一起发送给接收者. 这样一个混合加密方案既保证了系统的实用性,又不会降低整个加密方案的强度.

$$Ks+Ke(B)\qquad Kd(B)+Ks$$

目前较为有名的单密钥算法当数由 J. L. Massey 和 Lai Xuejia 发明的 IDEA 算法.



Kd[A]为 A 的私钥[解密用],Ke[A]为 A 的公钥[加密用];

Ke[B]与 Kd[B]分别为 B 的公钥与私钥;

Ks 为单密钥加密方案中所用的密钥;

加密方案 1 为 RSA 与单密钥加密的混合;

加密方案 2 为 RSA 加密,解密方案 1、2 分别与之对应.

图 2 Sem 加密方案

IDEA 的软件实现可以达到比 DES 更快的速度,且至今为止未被密码分析学家们找出弱点. 事实上甚至可以认为 IDEA 的强度比 RSA 要高,因为有结果表明,穷尽 IDEA 密码中可能的 128bit 密钥的计算量,大致相当于分解一个 3100 位的 RSA 密钥. Sem 目前采用的是由 IDEA 的作者之一 Massey 发明的另一分组加密算法,即 SAFER k-64. 可以认为这样一个混合加密方案的强度不低于纯粹 RSA 体制.

3. 签名与认证

为确保用户发出的邮件不被替换或窜改,可以给邮件进行签名. 用户 A 欲给 B 发送邮件时,首先对邮件进行变换以得到代表邮件的一个"缩影"(约 128bit);然后用自己的私钥对邮件"缩影"进行加密以产生所谓的签名. 这个签名连同邮件发送给用户 B 后,B 就可以用 A 的公钥检验签名的真实性. 图 2 给出了 Sem 的加密和解密及签名和认证过程. 图中的"变换"是一个单向 hash 函数. 它的作用类似于通常的求校验和或 CRC. 通过对邮件进行变换可以得到反映邮件信息的"缩影". 该 hash 变

换函数的设计使得攻击者不可能以较小的计算量伪造出具有相同"缩影"的不同报文.

系统公开"缩影"变换算法. 这样邮件的接收者可以针对解密后的邮件产生缩影,并与发送者签名所包含的缩影进行比较. 若二者相同,则表明 B 所收到的邮件确实是由 A 发出的.

密钥管理

在电子邮件安全系统 Sem 中,密钥管理占了相当的份量. Sem 的密钥管理主要分二类,即在用户端的"个人密钥管理系统",以及系统范围内的"公钥服务器"上的密钥管理.

1. 个人密钥管理

在 Sem 中,每个 e-mail 用户都有一或多个密钥箱. 一个密钥箱就是一个文件. 每个密钥箱中存放着一个一个的密钥记录. 每个密钥记录包括如下内容:

· 密钥实体(公钥或私钥)
· 密钥生成日期
· 密钥柄(Key Handle)
· 用户 ID(用户名字)

其中,密钥柄是公钥的最低 64bit,它用于在 Sem 内部检索公钥及相应的私钥. 用户所有的私钥存放在私钥箱中;用户收集的所有公钥存放在公钥箱中. 用户的私钥可以用口令进行保护. 这种保护可以在密钥箱一级上进行,也可以深入箱中对每一个私钥分别用口令保护.

用户 A 发送邮件给用户 B 时,选择 B 的一个公钥进行加密,并附上该公钥的密钥柄,以便 B 的 Sem 系统根据密钥柄自动选择正确的解密密钥. 用户 A 用于签名的私钥的密钥柄(实际上是相应公钥的低 64 位)也一并发给 B 以便他正确地认证该签名.

个人密钥管理的主要功能有:

① 密钥生成:包括 RSA 密钥的生成及传统加密算法临时会话密钥的生成;

② 密钥的添加/撤销;

③ 密钥箱的组织与维护;

④ 维持与公钥服务器的一致性.

2. 公钥服务器

为给 B 发送加密邮件,用户 A 需要有 B 的公钥. 这可以通过多种渠道得到:电话查询,信件邮寄,甚至公开出版物等. 但对于一个真正实用化、大型化的电子邮件系统而言,设置一个可信的公钥服务器,通过它来发布所有用户的公开密钥,应该是一种自然的选择. 假定用户 A 通过公钥服务器得到了用户 B 的公钥,那么这个公钥肯定就是属于 B 的. 也就是说,用户通过公钥服务器发布的公钥应该是未被窜改的. 这个所谓的"公钥可信性"是整个 RSA 体制中重要的一环.

事实上,在 Internet 上确实存在一些负责收集用户公钥的服务. 如著名的 PGP 公钥加密软件(由 Philip Zimmermann 等开发)的用户,便可以通过电子邮件向指定的公钥服务器(如设在 MIT 的一个公钥服务器的 e-mail 地址是 pgp-public-keys@pgp. mit. edu)注册或查询用户的公钥. 但这类公钥服务器目前均不保证公钥的可信性.

考虑公钥被窜改的情况. 假定攻击者窜改了服务器中保存的 B 的某个公钥 KB. 此后 A 得到的关于 B 的公钥 KB' 实际上是属于 C 的. A 用 KB' 加密邮件并发送给 B,希望只有 B 能阅读它. 但 C 截获邮件,并将之解密(因为 KB' 的解密密钥实际上是在 C 手里). C 然后对阅读并/或窜改后的邮件用真正属于 B 的公钥 KB 加密,并发送给 B. B 根据密钥柄选择正确的私钥解密. B 对于邮件已泄露一无所知.

为避免公钥被窜改,公钥服务器必须是可信的. 为此,服务器将给每一个由它发布的公钥附上它自己的签名. 这样,当用户第一次从服务器取得另一个用户的公钥时,可以利用公钥服务器本身的公开密钥来检验所附的签名,从而保证用户公钥的可信性. 服务器自身的密钥可以通过可信渠道(如正式出版物)发布.

当用户希望把自己的公钥登载到公钥服务器上时,也应先利用服务器的公钥进行加密,再发送给服务器.

总之,Sem 的公钥管理机制用一句话来说,就是在可信服务器集中控制下的管理.