§5.3 数字签名

数字签名是以密码学的方法对数据文件产生的一组代表签名者身份和数据完整性的数据信息。它提供了一种鉴别方法，以解决伪造、抵赖、冒充等问题。在身份认证、数据完整性、不可否认性以及匿名性等方面有重要应用，尤其是大型网络安全通信中的密钥分配、认证及电子商务系统中具有重要作用。

数字签名与手写签名的区别：手书签字是模拟的，它反映某个人的个性特征是不变的；数字签名是数字串，它随被签的对象而变化，即同一当事者对不同文件的数字签名是不相同的，某一个文件的数字签名不可能被直接复制到另一个文件上进行伪造签名。

§5.3 数字签名

数字签名是一种包括防止源点或终点否认的认证技术。可用来保护信息的真实性、完整性和信息的来源。

数字签名必须保证以下五点：

（1）可验证：签字是可以被确认的；

（2）防抵赖：发送者事后不承认发送报文并签名；

（3）防假冒：攻击者冒充发送者向收方发送文件；

（4）防篡改：收方对收到的文件进行篡改；

（5）防伪造：收方伪造对报文的签名。

§5.3 数字签名

数字签名技术是公钥密码体制的一种应用。签名者使用自己私钥对签名明文的“摘要”加密，就生成了该文件的“数字签名”。签名者将明文和数字签名一起发送给接收者，接收者用该签名者公布的公钥来解开数字签名，将其与明文的“摘要”进行比较，便可检验文件的真伪，并确定签名者的身份。

数字签名体制一般由签名算法和验证算法两部分组成。签名算法或签名密钥是秘密的，只有签名人掌握；验证算法应当公开，以便于他人进行验证。

§5.3 数字签名

对消息的签名与对消息的加密有所不同，消息加解密可能是一次性的，它只要求在解密之前是安全的；而一个签名的消息很可能在多年后仍需验证其签名，且可能需要多次验证此签名。

因此，签名的安全性和防伪造的要求更高些，且要求验证速度比签名速度要快些，尤其是对在线实时验证。

§5.3 数字签名

- **RSA数字签名体制**

算法描述：

（1）体制参数

令$n = pq$，选e并计算出d使$ed = 1 \bmod \phi(n)$，公开n和e，将p、q和d保密。

（2）签字过程

对消息m，用户A计算：$S = Sig_k(m) = m^d \bmod n$

s即为m的数字签名。

（3）验证过程

用户B收到m, s，验证$Ver_k(m, s)$=真$\Leftrightarrow m \equiv s^e \pmod n$

§5.3 数字签名

RSA签名体制的缺陷：

（1）任何人可以通过某个s计算： $m = s^e \bmod n$伪造用户对m的签名s

（2）若知道消息m_1、m_2的签名分别为s_1、s_2，则可伪造消息m_1m_2的签名$S = Sig_k(m_1m_2) = s_1s_2 \bmod n$

（3）签名速度慢。

通过增加单向散列函数可以克服上述缺陷。

§5.3 数字签名

➢ ElGamal数字签名体制

ElGamal数字签名方案的安全性主要是基于有限域上离散对数问题的难解性。

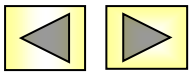
ElGamal数字签名体制的算法描述：

（1）体制参数

p为素数，$g \in F_p^*$是一个本原元，

随机选择整数x，$0 < x < p-1$，计算 $y = g^x \bmod p$

p、g为系统公开参数，x为私钥，y为公钥



§5.3 数字签名

（2）签名算法

设消息为m，用户A秘密选取随机数k， $0 < k < p-1$

并计算： $r= g^k \bmod p$

$s=(m-xr)k^{-1} \bmod (p-1)$

(r, s)即为m的数字签名。

（3）验证过程：

用户B收到m, r, s后，验证

$Ver(m, r, s)=真 \Leftrightarrow y^r r^s \equiv g^m \pmod p$

因为 $y^r r^s = g^{xr} g^{ks} = g^{xr+ks} \bmod p = g^m \bmod p$

ElGamal数字签名对同一消息m，由于随机数k不同而有不同的签名结果。

§5.3 数字签名

对ElGamal签名方案安全性的讨论：

若Oscar在不知道x的情况下企图伪造一个给定消息m的签名：

$Sig_{oscar}(m, k) = (r, s)$

（1）Oscar先选定一个s，然后企图找r，这样，他就必须解一个关于未知数r的方程： $y^r r^s \equiv g^m \pmod p$

这个方程是一个已知无可行解法的难处理问题！

（2）Oscar先选定一个r，使其满足： $y^r r^s \equiv g^m \pmod p$，

于是，$r^s \equiv y^{-r} g^m \pmod p$，这样他就必须计算离散对数 $\log_r(y^{-r} g^m)$，这自然是难处理的问题！

§5.3 数字签名

（3）若两者r, s都被 Oscar首先选定，然后企图解出一个随机消息m，使得$y^r r^s \equiv g^m \pmod p$，Oscar无法成功。

（4）Oscar同时选择r, s和m来伪造签名问题：

假设i和j是整数，$0 < i < p-1$，$0 < j < p-1$，且$(j, p-1) = 1$，先完成下列计算：

$$r = g^i y^j \pmod p$$

$$s = -r\, j^{-1} \pmod{p-1}$$

$$m = -r\, i\, j^{-1} \pmod{p-1}$$

(其中j^{-1}是用模p-1来计算的)

§5.3 数字签名

可以证实(r, s)是一个消息m的有效签名:

$$
\begin{aligned}
y^r r^s &\equiv y^{g^i y^j} (g^i y^j)^{-g^i y^j j^{-1}} \pmod p \\
&\equiv y^{g^i y^j} g^{-ij^{-1} g^i y^j} y^{-g^i y^j} \pmod p \\
&\equiv g^{-ij^{-1} g^i y^j} \pmod p \\
&\equiv g^{-ij^{-1} r} \pmod p \\
&\equiv g^m \pmod p
\end{aligned}
$$

§5.3 数字签名

➢ DSS数字签名标准

1991年8月由美国NIST公布，于1994年12月1日采纳为标准DSS。是在ElGamal方案基础上设计的。DSS中所采用的算法为DSA。

DSA算法描述：

（1）体制参数

p：素数且$2^{511}<p<2^{512}$；　　q：素数且$2^{159}<p<2^{150}$且q|p-1；

g：　$g = h^{(p-1)/q} \bmod p$，其中h是小于(p-1)且满足

$h^{(p-1)/q} \bmod p > 1$的任意数；

x：小于q的正整数，　　y：　$y = g^x \bmod p$

算法使用一个单向散列函数H(m)，标准指定为SHA算法。

其中p、q、g为系统公开参数，公钥为y，私钥为x

§5.3 数字签名

（2）签名算法

设消息为m(0 < m < p)，用户A随机选取小于q的随机数k，

并计算：　　$r = (g^k \bmod p) \bmod q$

$s = (H(m) + xr)k^{-1} \bmod q$

(r, s)即为m的数字签名。

（3）验证算法

用户B收到m, r, s后，计算

$$w = s^{-1} \bmod q, \quad u_1 = (H(m)w) \bmod q,$$

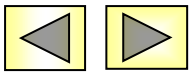
$$u_2 = (rw) \bmod q, \quad v = ((g^{u_1} \cdot y^{u_2}) \bmod p) \bmod q$$

§5.3 数字签名

如果v = r，则签名有效。

因为

$$
\begin{aligned}
v &= ((g^{u_1} \cdot y^{u_2}) \bmod p) \bmod q \\
&= ((g^{H(m)\cdot w \bmod q} \cdot y^{rw \bmod q}) \bmod p) \bmod q \\
&= (g^{H(m)\cdot w + xrw \bmod q} \bmod p) \bmod q \\
&= (g^{s^{-1}(H(m)+xr) \bmod q} \bmod p) \bmod q \\
&= (g^{s^{-1}sk \bmod q} \bmod p) \bmod q \\
&= (g^{k} \bmod p) \bmod q \\
&= r
\end{aligned}
$$



§5.3 数字签名

- 其他数字签名类型

群签名：允许群中各成员以群的名义匿名地签发消息。只有群的成员能代表这个群签名；签名接收者能验证它是这个群的合法签名，但不知具体是哪个成员；在发生争端时，借助群成员或可信机构能识别出那个签名者。用途：投标

盲签名：消息M的拥有者A，想让B对M签名，但又不希望B知道M的内容，而只是让B以证人身份证实在某一时刻存在文件M。

签名步骤：A将M乘以一个随机数(盲因子)得M'并发送给B;

B在M'上签名得SIG(M')，并发送给A;

A通过去除盲因子，由SIG(M')得到SIG(M)。

§5.3 数字签名

双重签名：实现三方通信时的身份认证和信息完整性、防抵赖的保护。应用：网上购物时，客户和商家之间要完成在线付款，在客户(甲) 、商家(乙)和银行(丙)之间将面临以下问题：甲向乙发送订单和甲的付款信息；乙收到订单后，要同丙交互，以实现资金转帐。但甲不愿让乙看到自己的帐户信息，也不愿让丙看到订购信息。此时甲使用双重签名技术对两种信息作数字签名，来完成以上功能。

§5.3 数字签名

双重数字签名的实现步骤如下：

1）甲分别对发给乙和丙的信息M_1、M_2生成摘要A_1、A_2；

2）甲把A_1、A_2合起来生成摘要A_3，并用私钥签名得$Sig(A_3)$；

3）甲把M_1、A_2和$Sig(A_3)$发给乙，把M_2、A_1和$Sig(A_3)$发给丙；

4）乙接收信息后，对M_1生成信息摘要A_1’，把A_1’和收到的A_2合起来并生成新的信息摘要，同时使用甲的公钥对A_3的签名进行验证，以确认信息发送者的身份和信息是否被修改过；

5）丙接收信息后，对M_2生成信息摘要A_2’，把A_2’和收到的A_1合起来并生成新的信息摘要，同时使用甲的公钥对A_3的签名进行验证，以确认信息发送者的身份和信息是否被修改过。

§5.3 数字签名

可仲裁的数字签名：

（1）单密钥加密方式仲裁者可以看见消息

签名过程：签名方A，接收方B，仲裁C，消息M

A→C：$M||Ek_{AC}[ID_A||H(M)]$

C→B：$Ek_{CB}\ [ID_A||M||Ek_{AC}[ID_A||H(M)]||T]$

说明：E：单钥加密，K_{AC}，K_{CB}分别为A和C、C和B的共享密钥。T：时戳，ID_A：A的身份，H(M)：M的杂凑值

B对收到的内容保存，以备争议时使用。由于B不知K_{AC}，不能直接验证A的签名，而是依靠仲裁者。

§5.3 数字签名

因此，仲裁者必须做到：

1）A相信C不会泄漏K_{AC}，不会伪造A的签名；

2）B相信C确实收到A的签名并验证无误后发送；

3）A和B都相信C可以公正地解决争议。

上述情况还隐含着C可以看到A给B的所有信息因而所有的窃听者也能看到。

§5.3 数字签名

（2）单密钥加密方式仲裁者不可以看见消息

A→C：$ID_A \| E_{k_{AB}}[M] \| E_{k_{AC}}[ID_A \| H(E_{k_{AB}}[M])]$

C→B：$E_{k_{CB}}[ID_A \| E_{k_{AB}}[M] \| E_{k_{AC}}[ID_A \| H(E_{k_{AB}}[M])] \| T]$

说明：K_{AB}为A和B的共享密钥。

在这种情况下C解密签名用散列函数验证消息，这时C只能验证消息的密文而不能读取其内容

方式一和二共同存在一个共性问题：

C和发送方A联手可以否认签名的信息，C和接收方B联手可以伪造发送方的签名。

§5.3 数字签名

（3）双密钥加密方式仲裁者不可以看见消息

A→C：$ID_A \| E_{Sk_A}[ID_A \| E_{Pk_B}(E_{Sk_A}[M])]$

C→B：$E_{Sk_C}[ID_A \| E_{Pk_B}(E_{Sk_A}[M]) \| T]$

在这种情况下，A对消息M双重加密，这种内部双重加密的消息对C以及对除B以外的其它人都是安全的，

本模式比上述两个模式具有以下好处：

1）在通信之前各方之间无须共享任何信息，避免了联手作弊；

2）即使SK_A暴露只要SK_C未暴露不会有错误标定日期的消息被发送；

3）从A发送给B的消息的内容对C和任何其他人是保密的。