



# 密码学

(第十一讲)

数字签名(1)

张焕国

武汉大学计算机学院

# 目 录

- 1、密码学的基本概念
- 2、古典密码
- 3、数据加密标准（DES）
- 4、高级数据加密标准（AES）
- 5、中国商用密码（SMS4）
- 6、分组密码的应用技术
- 7、序列密码
- 8、习题课：复习对称密码
- 9、公开密钥密码（1）

# 目 录

- 10、公开密钥密码 (2)
- 11、*数字签名 (1)*
- 12、数字签名 (2)
- 13、HASH函数
- 14、认证
- 15、密钥管理
- 16、PKI技术
- 17、习题课：复习公钥密码
- 18、总复习/检查：*综合实验*

# 一、数字签名的基本概念

在人们的工作和生活中，许多事物的处理需要当事者签名。

签名起到确认、核准、生效和负责任等多种作用。

签名是证明当事者的身份和数据真实性的一种信息。

签名可以用不同的形式来表示。

# 一、数字签名的基本概念

在传统的以书面文件为基础的事物处理中，采用书面签名的形式：

手签、印章、手印等

书面签名得到司法部门的支持。

在以计算机文件为基础的现代事物处理中，应采用电子形式的签名，即数字签名（*Digital Signature*）。

数字签名已得到一些国家的法律支持。

# 一、数字签名的基本概念

一种完善的签名应满足以下三个条件：

签名者事后不能抵赖自己的签名；

任何其他人不能伪造签名；

如果当事人的双方关于签名的真伪发生争执，能够在公正的仲裁者面前通过验证签名来确认其真伪。

# 一、数字签名的基本概念

数字签名基于密码技术，其形式是多种多样的：

通用签名、仲裁签名、不可否认签名、盲签名、群签名、门限签名等。

- 1994年月美国政府正式颁布了美国数字签名标准DSS (Digital Signature Standard)。
- 1995年我国也制定了自己的数字签名标准 (GB15851 - 1995)。
- 2004年我国颁布《中华人民共和国电子签名法》。

## 二、数字签名的模型

- 一个数字签名体制包括两个方面的处理：  
施加签名和验证签名。
- 设施加签名的算法为SIG，产生签名的密钥为K，被签名的数据为M，产生的签名信息为S，则有  
$$SIG(M, K) = S。$$

## 二、数字签名的模型

- 设验证签名的算法为  $VER$ ，用  $VER$  对签名  $S$  进行验证，可鉴别  $S$  的真假。即

$$VER(S, K) = \begin{cases} \text{真}, & \text{当 } S = SIG(M, K); \\ \text{假}, & \text{当 } S \neq SIG(M, K)。 \end{cases}$$

## 二、数字签名的模型

- 签名函数必须满足以下条件，否则文件内容及签名被篡改或冒充均无法发现：  
    当 $M' \neq M$ 时，有 $SIG(M', K) = SIG(M, K)$ 。
- 条件 要求签名 $S$ 至少和被签名的数据 $M$ 一样长。当 $M$ 较长时，应用很不方便。
- 将条件 改为：虽然当 $M' \neq M$ 时，存在 $S = S'$ ，但对于给定的 $M$ 或 $S$ ，要找出相应的 $M'$ 在计算上是不可能的。

## 二、数字签名的模型

签名  $S$  只能由签名者产生，否则别人便可伪造，于是签名者也就可以抵赖。

收信者可以验证签名 $S$ 的真伪。这使得当签名 $S$ 为假时收信者不致上当。

签名者也应有办法鉴别收信者所出示的签名是否是自己的签名。这就给签名者以自卫的能力。

## 三、利用公钥密码实现数字签名

### 1、一般方法：

- 对于一个公钥密码，如果满足

$$E(D(M, K_d), K_e) = M,$$

则可确保数据的真实性。

- 凡是能够确保数据的真实性的公开密钥密码都可用来实现数字签名，例如RSA密码、ELGamal密码、椭圆曲线密码ECC等都可以实现数字签名。

## 三、利用公钥密码实现数字签名

### 1、一般方法：

- 为了实现数字签名，应成立管理机构；  
制定规章制度，  
统一技术问题，  
用户登记注册，  
纠纷的仲裁等。

## 三、利用公钥密码实现数字签名

### 1、一般方法：

• 签名通信协议： $A \xrightarrow{M} B$

A用自己的解密密钥 $K_{dA}$ 对数据M进行签名：

$$S_A = D(M, K_{dA})$$

如果不需要保密，则A直接将 $S_A$ 发送给用户B。

如果需要保密，则A查到B的公开的加密钥 $K_{eB}$ ，并用 $K_{eB}$ 对 $S_A$ 再加密，得到密文C，

$$C = E(S_A, K_{eB})$$

最后，A把C发送给B，并将 $S_A$ 或C留底。

## 三、利用公钥密码实现数字签名

*B* 收到后，若是不保密通信，则先查到*A*的公开加密钥 $K_{eA}$ ，然后用 $K_{eA}$ 对签名进行验证：

$$E(S_A, K_{eA}) = E(D(M, K_{dA}), K_{eA}) = M$$

若是保密通信，则*B*先用自己的保密的解密密钥 $K_{dB}$ 对*C*解密，然后再查到*A*的公开加密钥 $K_{eA}$ ，用 $K_{eA}$ 对签名进行验证：

$$D(C, K_{dB}) = D(E(S_A, K_{eB}), K_{dB}) = S_A$$

$$E(S_A, K_{eA}) = E(D(M, K_{dA}), K_{eA}) = M$$

如果能够恢复出正确的*M*，则说明 $S_A$ 是*A*的签名，否则 $S_A$ 不是*A*的签名。

## 三、利用公钥密码实现数字签名

### 1、一般方法：

- 签名通信协议安全分析：

因为只有A才拥有 $K_{dA}$ ，而且由公开的 $K_{eA}$ 在计算上不能求出保密的解密密钥 $K_{dA}$ 。因此签名的操作只有A才能进行，任何其他人都不能进行。所以， $K_{dA}$ 就相当于A的印章或指纹，而 $S_A$ 就是A对M的签名。对此A不能抵赖，任何其他人不能伪造。

## 三、利用公钥密码实现数字签名

### 1、一般方法：

- 签名通信协议安全分析：

事后如果A和B关于签名的真伪发生争执，则他们应向公正的仲裁者出示留底的签名数据，由仲裁者当众验证签名，解决纠纷。

## 三、利用公钥密码实现数字签名

### 1、一般方法：

- **签名通信协议的问题：**

*验证签名的过程就是恢复明文的过程。而B事先并不知道明文M，否则就用不着通信了。那末B怎样判定恢复出的M是否正确呢？*

*怎样阻止B或A用A以前发给B的签名数据，或用A发给其他人的签名数据来冒充当前A发给B的签名数据呢？*

- **仅仅靠签名本身并不能解决这些问题。**

## 三、利用公钥密码实现数字签名

### 1、一般方法：

- 解决问题的一种办法：  
合理设计明文的数据格式：

发方标识	收方标识	报文序号	时间	数据	纠错码
------	------	------	----	----	-----

$M = \langle A, B, I, T, DATA, CRC \rangle$

记其中  $H = \langle A, B, I, T \rangle$ 。

于是，A以  $\langle H, \text{SIG}(M, K_{dA}) \rangle$  为最终报文发给B，  
其中H为明文形式。

## 三、利用公钥密码实现数字签名

### 1、一般方法：

- 解决问题的一种办法：

只要用A的公钥验证签名并恢复出正确的附加信息  $H = \langle A, B, I, T \rangle$ ，便可断定明文M是否正确。

设附加信息  $H = \langle A, B, I, T \rangle$  的二进制长度为  $l$ ，则错判概率

$$p_e = 2^{-l}。$$

## 三、利用公钥密码实现数字签名

### 1、一般方法：

- 解决问题的另一种办法：对HASH (M) 签名，而不直接对M签名。

发方标识	收方标识	报文序号	时间	数据	HASH码
------	------	------	----	----	-------

$M = \langle A, B, I, T, DATA \rangle$

$S = \text{SIG}(\text{HASH}(M), K_{dA})$

传输格式： $\langle M, S \rangle$

## 三、利用公钥密码实现数字签名

### 2、利用RSA密码实现数字签名：

- *对于RSA密码*

$$D(E(M)) = (M^e)^d = M^{ed} = (M^d)^e = E(D(M)) \pmod{n},$$

*所以RSA可同时确保数据的秘密性和真实性。*

- *因此利用RSA密码可以同时实现数字签名和数据加密。*

## 三、利用公钥密码实现数字签名

### 2、利用RSA密码实现数字签名：

- 设 $M$ 为明文， $K_{eA} = \langle e, n \rangle$ 是 $A$ 的公开钥， $K_{dA} = \langle d, p, q, (n) \rangle$ 是 $A$ 的保密的私钥，则 $A$ 对 $M$ 的签名过程是，

$$S_A = D(M, K_{dA}) = (M^d) \bmod n$$

$S_A$ 便是 $A$ 对 $M$ 的签名。

- 验证签名的过程是，

$$E(S_A, K_{eA}) = (M^d)^e \bmod n = M$$

## 三、利用公钥密码实现数字签名

### 3、对RSA数字签名的攻击

一般攻击：

- 设 $e$ 和 $n$ 是用户A的公开密钥，所以任何人都可以获得并使用 $e$ 和 $n$ 。攻击者可随意选择一个数据 $Y$ ，并用A的公钥计算

$$X = (Y)^e \bmod n$$

- 因为  $Y = (X)^d \bmod n$ ，于是可以用 $Y$ 伪造A的签名。因为 $Y$ 是A对 $X$ 的一个有效签名。
- 注意：这样的 $X$ 往往无正确语义！

## 三、利用公钥密码实现数字签名

### 3、对RSA数字签名的攻击

利用已有的签名进行攻击：

- 攻击者选择随机数据  $M_3$ ，且  $M_3 = M_1 M_2 \pmod n$ 。
- 攻击者设法让A对  $M_1$  和  $M_2$  签名：  
$$S_1 = (M_1)^d \pmod n, \quad S_2 = (M_2)^d \pmod n$$
- 于是可以由  $S_1$  和  $S_2$  计算出A对  $M_3$  的签名。因为  
$$(S_1 S_2) = (M_1)^d (M_2)^d \pmod n = (M_3)^d \pmod n = S_3$$
- 对策：A不对数据M签名，而是对  $\text{HASH}(M)$  签名。

## 三、利用公钥密码实现数字签名

### 3、对RSA数字签名的攻击

利用已有的签名进行攻击：

- 此时：

$$S_1 = \text{HASH}(M_1)^d \bmod n, \quad S_2 = \text{HASH}(M_2)^d \bmod n$$

而， $\text{HASH}(M_1)^d \text{HASH}(M_2)^d = \text{HASH}(M_1M_2)^d \bmod n$

- 所以： $S_3 = S_1S_2$

- 于是不能由 $S_1$ 和 $S_2$ 计算出A对 $M_3$ 的签名。

## 三、利用公钥密码实现数字签名

### 3、对RSA数字签名的攻击

攻击签名获得明文:

- 攻击者截获 $C$ ,  $C = (M)^e \bmod n$ 。
- 攻击者选择小的随机数 $r$ , 计算:  $x = r^e \bmod n$   
 $y = xC \bmod n$ ,  $t = r^{-1} \bmod n$
- 攻击者让A对 $y$ 签名, 于是攻击者又获得:  
 $S = y^d \bmod n$
- 攻击者计算  $tS = r^{-1}y^d = r^{-1}x^d C^d = C^d = M \bmod n$
- **对策: A不对数据 $M$ 签名, 而是对 $\text{HASH}(M)$ 签名。**

# 三、利用公钥密码实现数字签名

## 3、对RSA数字签名的攻击

结论：

- 不对数据 $M$ 签名，而是对 $HASH(M)$ 签名。
- 使用时间戳
- 对于同时确保秘密性和真实性的通信，应当先签名后加密。

## 三、利用公钥密码实现数字签名

### 4、RSA 数字签名的应用：PGP

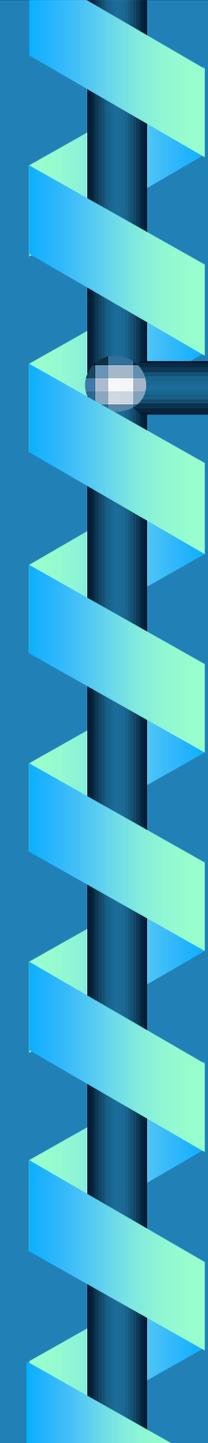
- 数据M经MD5处理
- 利用RSA对HASH(M)签名, 得到 S
- 使用ZIP对<M, S>压缩
- 再用IDEA对压缩数据加密： $IDEA(ZIP(M, S))$
- 用RSA对IDEA的密钥加密： $RSA(k)$
- 形成数据： $\langle IDEA(ZIP(M, S)), RSA(k) \rangle$
- 将数据转换成ASCII码。

# 习题

为什么数字签名能够确保真实性？

说明对于RSA的数字签名，为什么先加密后签名不安全？

说明HASH函数在RSA数字签名中的作用。



---

谢谢！