

## 对称密码技术（MAC 算法）的局限性

只能实现	数据完整性	什么是 否认
不能实现	非否认	耍赖，不承认曾经参与过某次通信
Q: 为什么对称密码技术不能实现非否认?		
<ul style="list-style-type: none"><li>因为双方都持有相同的密钥（信息是对称的）</li><li>接收方可以产生相同的消息，所以发送方可以诬赖消息是接收方伪造的</li></ul>		

1. 只能实现数据完整性，不能实现非否认
2. 什么是否认？耍赖，不承认曾经参加过某次通信
- 3 Q: 为什么对称密码技术不能实现非否认？
  - 因为双方都持有相同的密钥（信息是对称的）
    - 接收方可以产生相同的消息，所以发送方可以诬赖消息是接收方伪造的

## 数字签名是基于公钥思想的数据完整性技术

数字签名：利用电子手段对电子文档进行签名，数字签名至少要满足手写签名的两个基本性质

- ① 别人可以 验证 数字签名
- ② 其他人 很难模仿 数字签名

数字签名的基本思想：

1. 发送者利用 自己的私钥 SK 产生消息的认证码 (类似于 MAC)
2. 只有发送者掌握 SK，所以该认证码只有发送者才能产生
3. 任何人都可以用相应的公钥 PK 验证认证码的合法性
4. 只要通过验证，就可以确信发送者产生了该消息
5. 认证码相当于发送者在消息上做的“签名”，故而称作 数字签名

总结：发送者用自己的私钥生成认证码，任何人都可以用公钥 pk 验证认证码的合法性，只要通过验证，就可以确定发送者产生了该消息

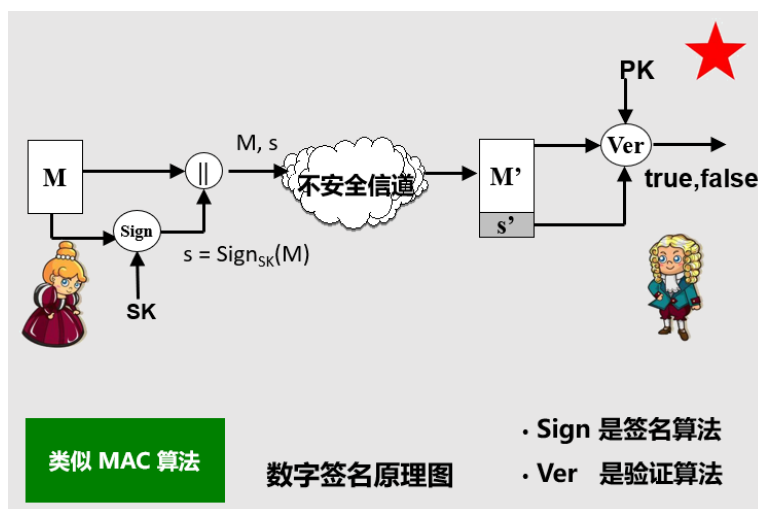
数字签名方案包括三个组成部分：

密钥生成 Setup：产生公钥/私钥

签名算法 Sign：利用私钥对消息产生数字签名

验证算法 Ver：利用公钥对数字签名进行验证

Setup	$\text{Setup}(1^k) = \text{PK} / \text{SK}$
签名	$\text{Sign}_{\text{SK}}(M) = s$
验证	$\text{Ver}_{\text{PK}}(M, s) = \text{true/false}$
$\text{Ver}_{\text{PK}}(M, \text{Sign}_{\text{SK}}(M)) = \text{true/false}$	



### 数字签名的主要安全特性

1. 抗伪造：任何人都不能伪造他的签名
2. 防篡改：任何人无法篡改已签名的消息
3. 非否认：签名者事后无法否认自己的签名。
4. 消息认证：接收者可以确信消息发送者的身份

### 抗伪造是数字签名的核心安全性要求

Q 什么叫 伪造签名？

A 在不知道私钥 SK 的情况下，产生签名 s，使得

$$\text{Ver}_{\text{PK}}(M, s) = \text{true}$$

则称 s 是对 M 伪造的签名 (PK 是 SK 对应的公钥)

也即 找到一对能通过验证算法校验的 (M, s)

Q 什么叫 抗伪造？

给定消息 M，在不知道私钥 SK 的情况下，产生签名 s，使得  $\text{Ver}_{\text{PK}}(M, s) = \text{true}$  是计算上不可行的 (就算已知一大堆消息以及其对应的签名也是如此)

## 数字签名 安全模型

- ① 完全攻破：攻击者能找到私钥 SK  
(UB: Unbreakability, 不可完全攻破)
- ① 泛伪造：攻击者可以对任何消息产生合法的签名  
(UU: Universal Unforgeability, 泛不可伪造)
- ① 选择性伪造：对别人选择的消息，攻击者能以不可忽略的概率产生一个合法的签名  
(SU: Selective Unforgeability, 选择性不可伪造)
- ④ 存在性伪造：攻击者能至少为一条消息产生合法的签名  
(EU: Existential Unforgeability, 不可存在性伪造)

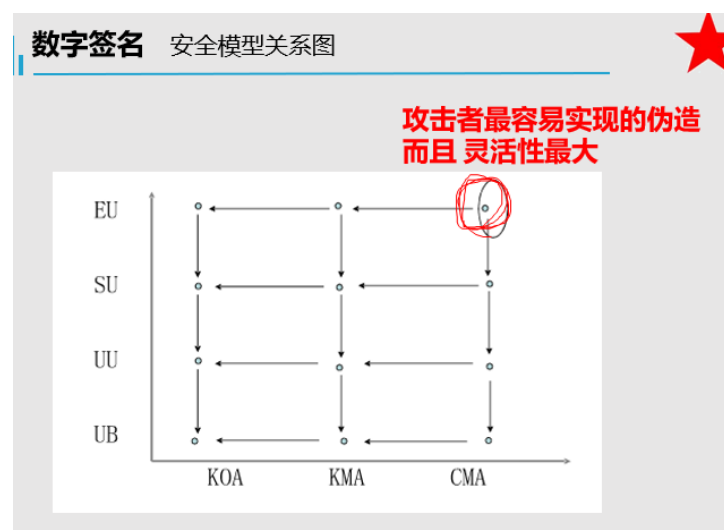
攻击者最容易实现的伪造(存在性伪造)

## 数字签名 攻击分类

依据攻击者获得信息的多少，对数字签名的攻击分类：

- ① 唯密钥攻击：攻击者只知道公钥  
(KOA: Key only attack)
- ② 已知消息攻击：攻击者拥有一系列用私钥签过的消息和相应的签名  
(KMA: Known message attack)
- ③ 选择消息攻击：攻击者任意选择一系列消息，并可获得相应的签名  
(CMA: Chosen message attack)

攻击者的目标----恢复私钥(但很难实现)，或伪造签名



数字签名方案的最高安全性要求：  
选择消息攻击下 不可存在性伪造 (EU-CMA)

## RSA 签名方案 描述

### ① 系统建立:

- 随机选择大素数 $p$ 、 $q$ , 计算 $n = pq$
- 随机选取 $e < \phi(n)$ , 且 $\gcd(e, \phi(n)) = 1$
- 计算 $d$ , 使 $ed \equiv 1 \pmod{\phi(n)}$
- $(e, n)$ 为公钥
- $d$ 为私钥

(与RSA加密方案的系统建立过程完全一样)

### ② 签名: $s = m^d \bmod n$ , $m \in \mathbb{Z}_n^*$

### ③ 校验: $m \stackrel{?}{=} s^e \bmod n$

## RSA 签名方案 存在性伪造

### 先看一下数字签名的原理

- 给定消息 $m$ , 计算签名  $s = \text{Sig}_{sk}(m)$
- 给定  $m$  和  $s$ , 验证合法性  $\text{Ver}_{pk}(m, s) = \text{true}$

### 存在性伪造的原理: 将产生签名的思路反过来

- 先选择 $s$ , 再构造相应的消息 $m$ , 使得  $\text{Ver}_{pk}(m, s) = \text{true}$
- 这样不知道私钥 $sk$ , 也可以产生满足验证算法的消息和签名, 这种伪造称为 存在性伪造

## 伪造的过程

- 攻击者随机选择  $s$
- 用签名者公钥 $(e, n)$ 计算  $M \leftarrow s^e \bmod n$
- 将 $(M, s)$ 作为消息/签名输出

Q:  $(M, s)$ 能否通过验证算法的校验呢?

- 很明显,  $(M, s)$  满足验证算法  $M = s^e \bmod n$  的要求
- 因此,  $(M, s)$  是一对合法的伪造

## RSA 签名方案 其他漏洞举例

- 可以利用两个消息的签名，产生新消息的签名
  - 如果攻击者知道消息  $m_1$  和  $m_2$  的签名，设分别是  $s_1 = m_1^d \bmod n$ ,  $s_2 = m_2^d \bmod n$   
则可以伪造消息  $m = m_1 m_2$  的签名  $s = s_1 s_2$
  - 因为 RSA 签名方案存在以下性质：  
 $(m_1 m_2)^d \equiv m_1^d m_2^d \pmod{n}$   
所以， $s^e \equiv (s_1 s_2)^e \equiv (m_1^d m_2^d)^e$   
 $\equiv ((m_1 m_2)^d)^e \equiv m_1 m_2$   
 $\equiv m \pmod{n}$

## HASH 函数在数字签名中的重要作用

- 重要作用
  - ① 抵抗存在性伪造
  - ② 加快计算速度
- 方法
  - ① “对消息  $m$  签名” 改为 “对  $H(m)$  签名”
- 原理
  - ① 抵抗存在性伪造：利用 Hash 函数的单向、抗碰撞等性质
  - ② 加快计算速度： $H(m)$  比  $m$  短得多

## 利用 Hash 函数改进 RSA 签名方案

- 改进后的方案
  - 签名： $s = H(m)^d \bmod n$
  - 验证： $H(m) \stackrel{?}{=} s^e \bmod n$
- 可抵抗“存在性伪造”
  - 攻击者随机选择  $s$ ，用签名者公钥计算  $h = s^e \bmod n$
  - 但计算一个  $m$ ，使得  $H(m) = h$  在计算上不可行
- 可抵抗“利用两个消息的签名，产生新消息的签名”
  - 因为  $(H(m_1)H(m_2))^d \neq H(m_1 m_2)^d \pmod{n}$

## 7.3 ElGamal 签名方案

- ① 系统建立
  - 随机选择大素数  $p$ ，及生成元  $g \in \mathbb{Z}_p^*$
  - 随机选取  $0 < x \leq p-2$ ，计算  $y = g^x \bmod p$
  - 公钥是  $(p, g, y)$
  - 私钥是  $x$(与 ElGamal 加密方案的系统建立过程完全一样)

目前， $p$  的长度至少应为 512 比特

- ② 签名
  - 对消息  $m$ ，随机选择  $0 < r \leq p-2$ ，然后计算：  
 $u = g^r \bmod p$

$$s = r^{-1}(m - xu) \bmod (p-1)$$

m 的签名为(u,s)

### ③ 验证

对于消息/签名 (m,(u,s)), 如果 :

$$y^u u^s \equiv g^m \pmod{p}$$

则(u,s)是 m 的有效签名

## ElGamal 签名方案 描述

### • 正确性:

$$y^u u^s = g^{xu} g^{rs}$$

$$= g^{xu+rs}$$

$$= g^m \bmod p$$

$$y = g^x, u = g^r$$

$$\begin{aligned} xu + rs &= xu + r(r^{-1}(m - xu)) \\ &= xu + m - xu \\ &= m \end{aligned}$$

## ElGamal 签名方案 存在性伪造

### ElGamal 签名方案 存在性伪造

#### • 对ElGamal签名的存在性伪造

- 攻击者随机选择  $0 < r, v \leq p-2$ , 且  $\gcd(v, p-1) = 1$

- 计算  $u = g^r y^v \bmod p$

$$s = -uv^{-1} \bmod (p-1)$$

$$m = -ruv^{-1} \bmod (p-1)$$

则(u,s)是对伪造的消息m的有效签名

#### • 伪造的正确性

$$y^u u^s = y^u (g^r y^v)^s = g^{xu} (g^r g^{xv})^s$$

$$= g^{xu+rs+xvs}$$

$$= g^m \bmod p$$

$$\begin{aligned} xu + rs + xvs &= xu + rs + xv(-uv^{-1}) \\ &= xu + rs - ux \\ &= rs \\ &= r(-uv^{-1}) \\ &= -ruv^{-1} \\ &= m \end{aligned}$$

## ElGamal 签名方案 利用 HASH 函数改进

- **签名**

$$u = g^r \bmod p,$$

$$s = r^{-1} (H(m, u) - x u) \bmod (p-1)$$

其中Hash函数  $H: \{0,1\}^* \rightarrow Z_p$

则m的签名为(u,s)

- **验证**

对于消息/签名(m,(u,s)), 如果:

$$y^u u^s = g^{H(m,u)} \bmod p$$

则(u,s)是m的有效签名。

## ElGamal 签名方案

### 注意

- ① 每次签名时, 必须选择不同的  $r$ , 否则私钥  $x$  可能会泄露
- ② 效率不如 RSA 签名高, 而且数据长度有扩张
- ③ 有很多种变形
- ④ 有的 ElGamal 签名方案的变形能在 某些假设下 被证明在 选择消息攻击下 是安全的

## ElGamal 签名方案 DSA

DSA (数字签名算法) 是 NIST 在 1991 年选定的数字签名标准

DSA 类似于 ElGamal, 但具有明显的优势

- ① 效率更高
- ② 签名更短

## 公钥基础设施(PKI) 概述

Q 通过网络进行交流和商业活动, 面临的最大问题是?

- 如何建立相互间的信任关系
- 如何确保信息的真实性、完整性、机密性和非否认

PKI 是解决这一系列问题的技术基础, 它是电子商务、电子政务的关键和基础技术

### 目的

- 为了提供可信任的高效密钥和证书管理, 以支持众多依赖于公钥的安全技术

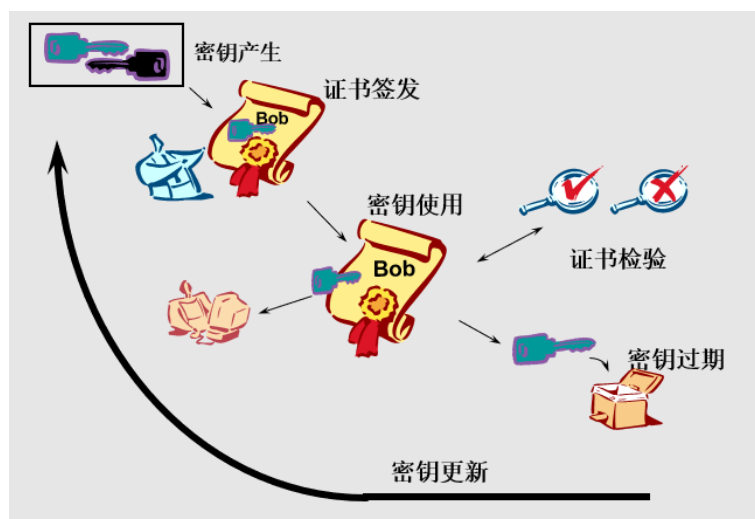
## PKI 的定义

一个使用公钥概念和密码技术实施和提供安全服务的具有普适性的安全基础设施的总称

PKI 不是特指某一个密码设备和管理设施，它是

- 生成、管理、存储、颁发和撤销公钥证书所需要的软硬件、人员、策略和规程的总和
- 提供密钥管理和数字签名服务的平台

## 公钥基础设施(PKI) 密钥生成周期



## PKI 相关标准 X.509 证书格式

- 版本 1、2、3
- 序列号在 CA 内部唯一
- 签名算法标识符指该证书中的签名算法
- 签发人名字 CA 的名字
- 有效时间: 起始和终止时间
- 实体名字

## PKI 相关产品

- ① VeriSign ([www.verisign.com](http://www.verisign.com))
  - 最大的公共 CA，最早推广 PKI 的公司之一
  - 最可信的公共 CA 之一，开发了一些使用工具
- ② Entrust 公司——世界一流
  - 产品：Entrust/PKI 5.0
  - 优点：
    - ① 管理和安全做得都很出色
    - ② 与其它产品兼容
    - ③ 支持各种标准:X.509 等