电子科技大学

University of Electronic Science and Technology of China

第五章

消息认证与数字签名

密码学应用

- 信息加密
- 消息认证
- 数字签名
- 身份认证

5.1 消息认证

消息（报文）认证

- 消息接收者验证消息来源真实性和消息完整性的过程，
  - 真实性：发送者真实非假冒——信源鉴别；
  - 完整性：消息在传送或存储过程中没被篡改、重放、乱序或延迟等；
- 防止主动攻击重要技术：
  - 假冒：
    - 冒充某合法实体发送一个消息
  - 内容修改：
    - 对消息内容篡改，包括插入、删除、转换和修改。
  - 顺序修改：
    - 对消息顺序修改，包括插入、删除和重新排序。
  - 计时修改：
    - 对消息延迟和重放

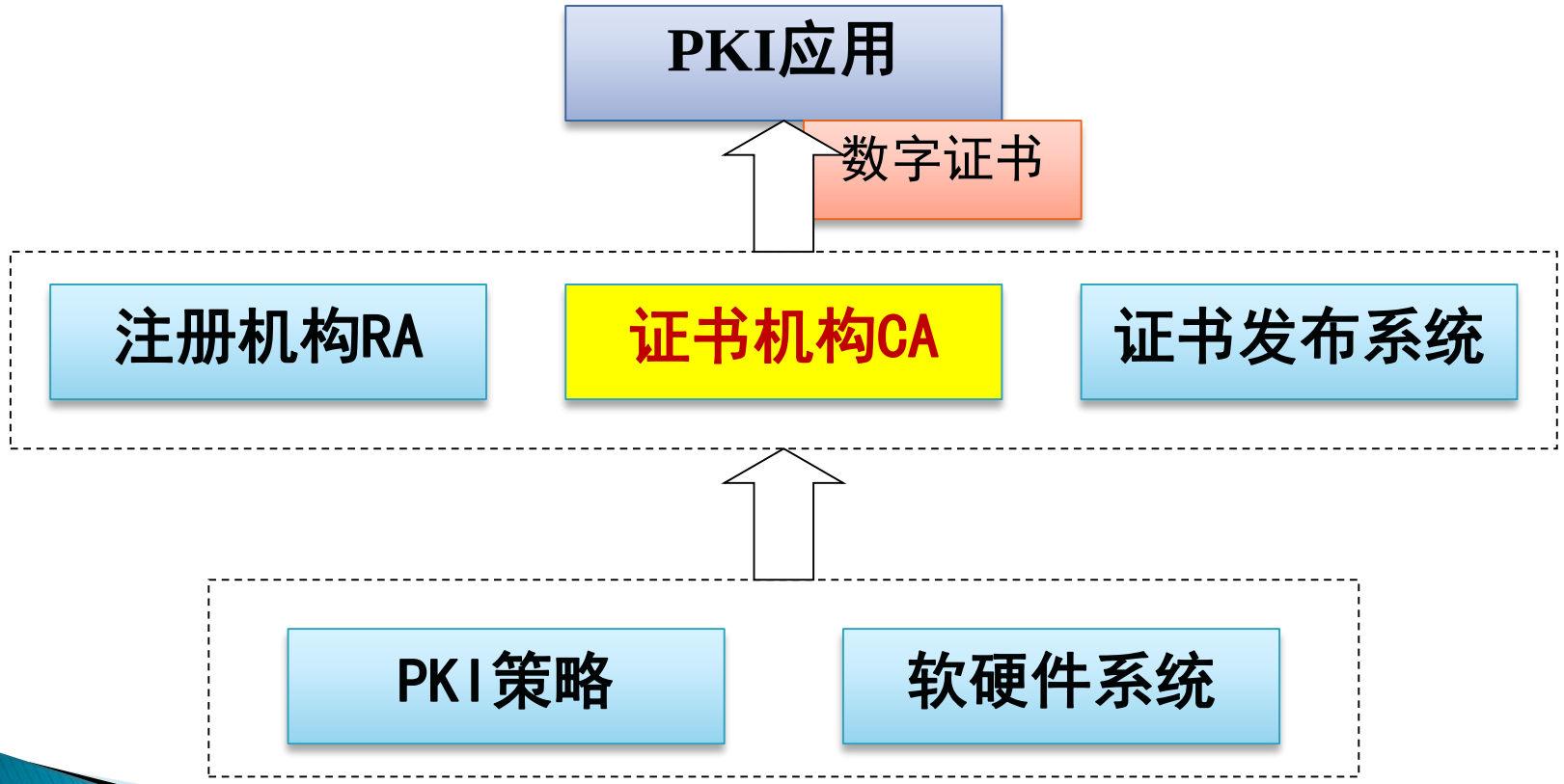
温故而知新——公钥管理解决方案

- 将公钥与身份绑定
  - 数字（公钥）证书
- 由可信第三方做担保
  - 权威机构（CA）管理、签名（盖章）、颁发
- 其他用户验证证书
  - 验证签名

温故而知新——公钥证书形式

- $C_A=[T, ID_A, KU_A, ID_{CA}]||Sig_{CA}$
  - 时间戳T保证证书时效性，防止重放旧证书
- $Sig_{CA}=D_{KRCA}(H)$
  - *签名一般使用方式：m//sig(m)*
- $H=hash([T, ID_A, KU_A, ID_{CA}])$
- 证书(签名)验证
  - $H=E_{KUCA}(Sig_{CA})$
  - $H`=hash([T, ID_A, KU_A, ID_{CA}])$
  - $H=?H`$
  - CA公钥获取：CA证书

温故而知新—— PKI的逻辑结构

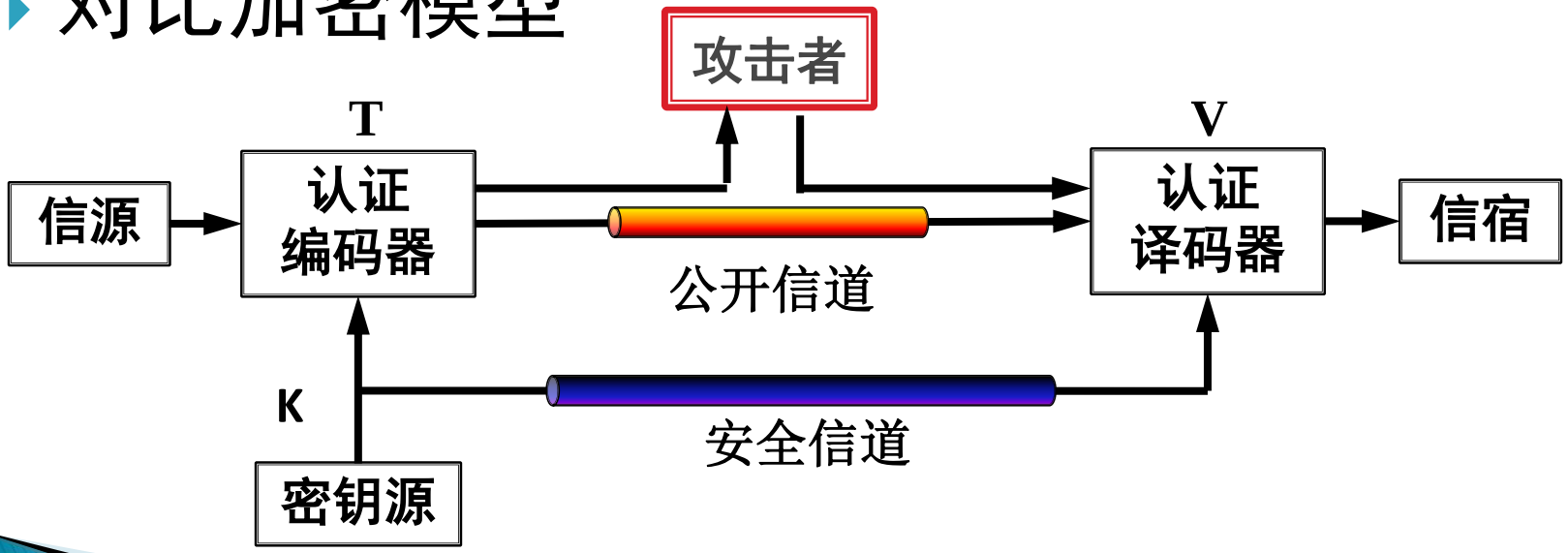


消息（报文）认证

- 核心思路：
  - 比较发送方发送的消息M和接收方收到的消息M`是否一致
  - M`＝？ M
- 类比快递、外卖
  - 如何保障外卖的完整性？

消息认证模型

- 三元组（K,T,V)
  - 密钥生成算法K
  - 标签算法T
  - 验证算法V
- 对比加密模型



认证（鉴别）函数

- 认证编码器、译码器抽象为认证函数
  - 发送方产生一个**认证标识**（Authentication Identification）
  - 给出合理**认证协议**(Authentication Protocol)
  - 接收者完成消息的鉴别（Authentication）

认证函数

- 分三类：
  - 消息加密函数(Message encryption)
    - 用完整信息的密文作为对信息的认证。
  - 消息认证码MAC(Message Authentication Code)
    - 对信源信息的一个编码函数
    - 公开函数+密钥产生一个固定长度的值作为认证标识
  - 散列函数(Hash Function)
    - 数字指纹（公开的函数），它将任意长的信息映射成一个固定长度的信息。

消息加密函数

- 消息自身加密作为认证度量
  - 用完整信息的密文作为对信息的认证。
  - 消息发送/接收方事先约定密钥
    - 信源：发送M+C，其中$C=E_K(M)$，认证标识
    - 信宿：接收M`+C，
    - 验证：$M= D_K(C)$，M`=?M

缺点：认证标识（完整密文）与消息等长，传输开销倍增

认证函数：消息认证码（MAC）

- 假定通信双方共享密钥K
  - 发送方使用K生成一个**固定大小**的短数据块，并将该数据块附加到消息后面

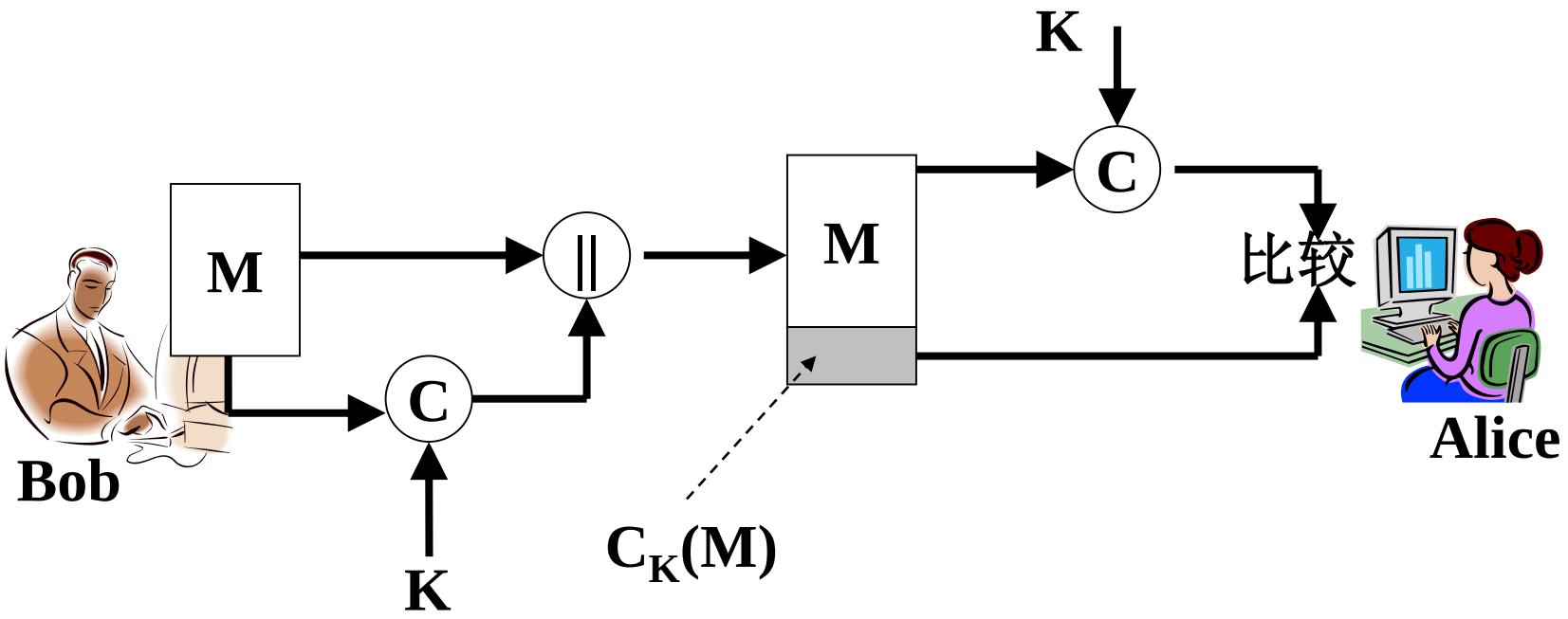
    $MAC=C_k(M)$

    send：M+MAC
  - 接收方接收到消息M`+MAC，使用K生成

    $MAC`=C_k(M`)$

    MAC`=？MAC
- MAC函数类似于加密函数，但固定大小
  - 不需要可逆性，因此在数学上比加密算法被攻击的弱点要少

MAC基本用法：消息认证

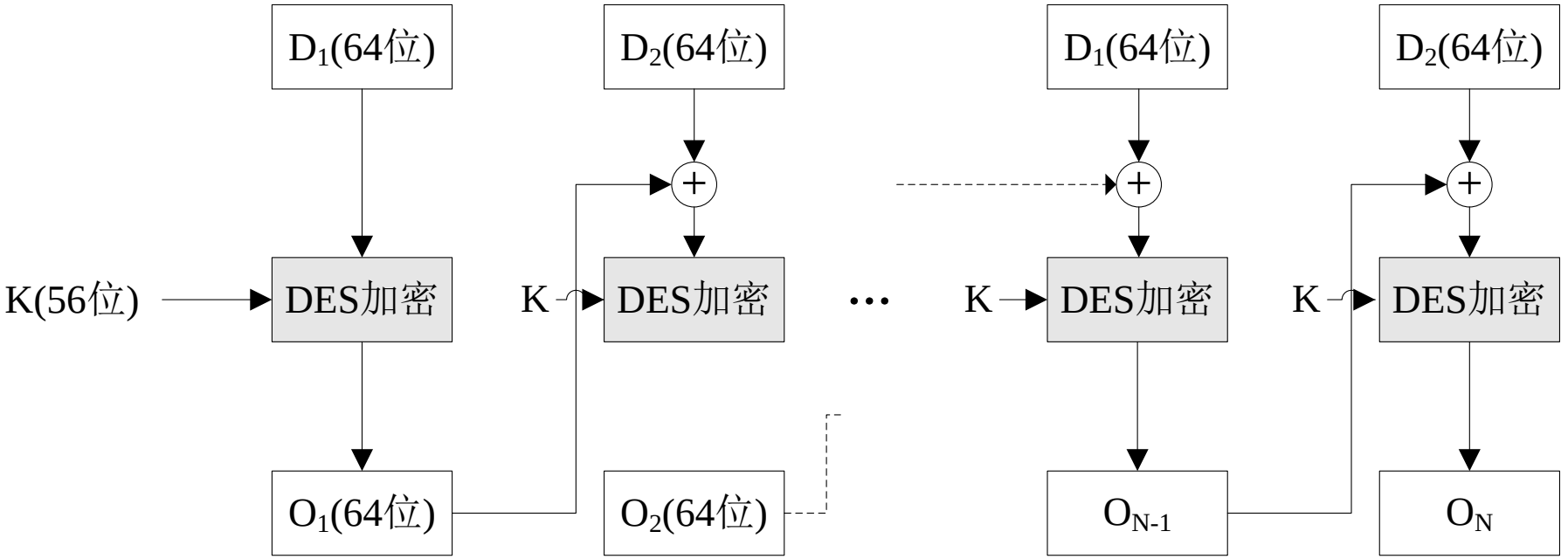


仅认证不保密

MAC安全要求

- MAC中使用了密钥，这点和对称密钥加密一样，如果密钥泄漏了或者被攻击了，则MAC的安全性则无法保证。

基于DES的消息认证码



MAC优缺点

- 优点：认证标识（码）大小固定且短
- 缺点：需要密钥，不需要从MAC解密出m
- 有没有不需密钥就能生成定长且短的认证标识的方法呢？

HASH

散列函数Hash Function

- 消息摘要、哈希函数、数字指纹、杂凑函数

$$h = H(M)$$

- 输入：任意长度的消息M
- 输出：一个固定短长度散列值H(M)
- 单向函数：
  - 正向计算容易，反向计算困难
- 不同消息不同指纹，用作消息标识
  - 消息M的所有位的函数：消息中任何一位或多位变化都将导致该散列值的变化。

安全HASH函数要求

- 任意长度数据块，产生固定长度散列值；
- 单向性：
  - 任意给定m ,计算H(m)相对容易；
  - 对任意给定h，找到m满足H(m)=h在计算上不可行；
- 安全性，冲突（碰撞）一定存在，但发现困难
  - 任意给定消息m1，找到m2≠m1满足H(m2)=H(m1)计算上不可行；
  - 找到任意消息对(m1,m2)，满足H(m1) = H(m2)计算上不可行。

简单的哈希算法

- 输入消息序列，以迭代的方式每次处理一个分组。
- 一个最简单的哈希函数：每个分组按比特异或(XOR)。
- 将消息M分成N个定长分组：
  - $M_1\ M_2\ M_3\ M_4\ \ldots M_N$
  - $H(M)=M_1 \oplus M_2 \oplus M_3 \oplus M_4 \oplus \ldots \oplus M_N$

Hash函数的分类——根据安全水平：

- 弱无碰撞
  - 对给定消息$x \in X$，在计算上几乎找不到异于x的$x' \in X$使$h(x)=h(x')$。
- 强无碰撞
  - 在计算上几乎不可能找到相异的x，x'使得$h(x)=h(x')$。
- 注：强无碰撞自然含弱无碰撞！

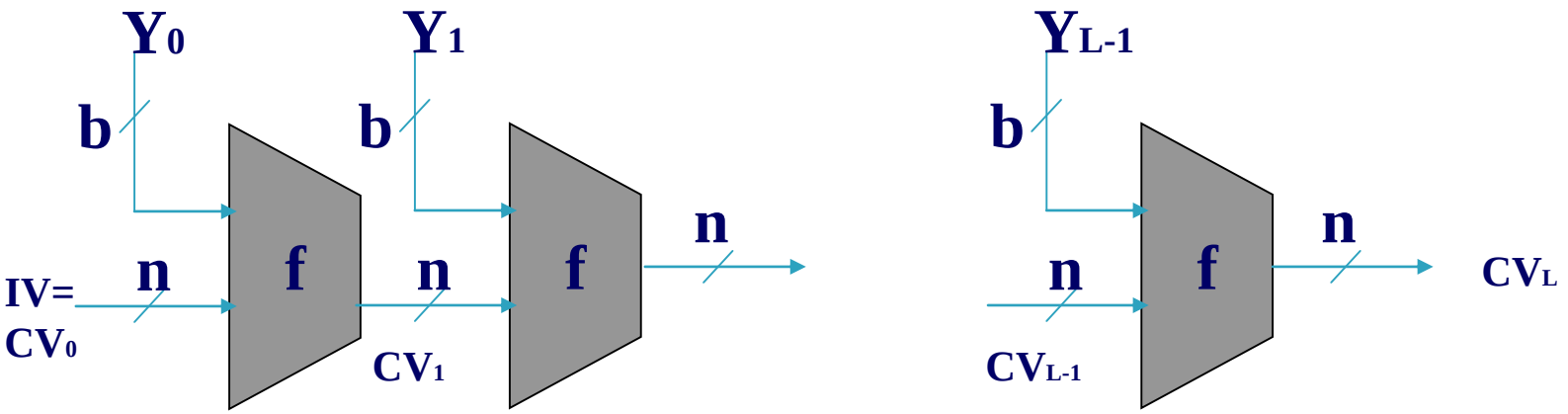
Hash函数的构造

- 基于数学难题
  - 计算速度慢，不实用
- 利用对称密码体制
- 直接设计

hash函数通用结构

- 几乎被所有hash函数使用
  - 把原始消息M分成一些固定长度的块Y_i
  - 最后一块填充
  - 设定初始值CV_0
  - 压缩函数f, $CV_i=f(CV_{i-1},Y_{i-1})$
  - 最后一个CV_i为hash值

hash函数通用结构



CV_0=IV= initial n-bit value

$CV_i = f(CV_{i-1}, Y_{i-1})$ $(1 \le i \le L)$

$H(M) = CV_L$

IV = initial value 初始值

CV = chaining value 链接值

Yi = ith input block (第i 个输入数据块)

f = compression algorithm (压缩算法）

n = length of hash code (散列码的长度)

b = length of input block(输入块的长度)

几种常用的HASH算法

- MD5
- SHA-1
- RIPEMD-160

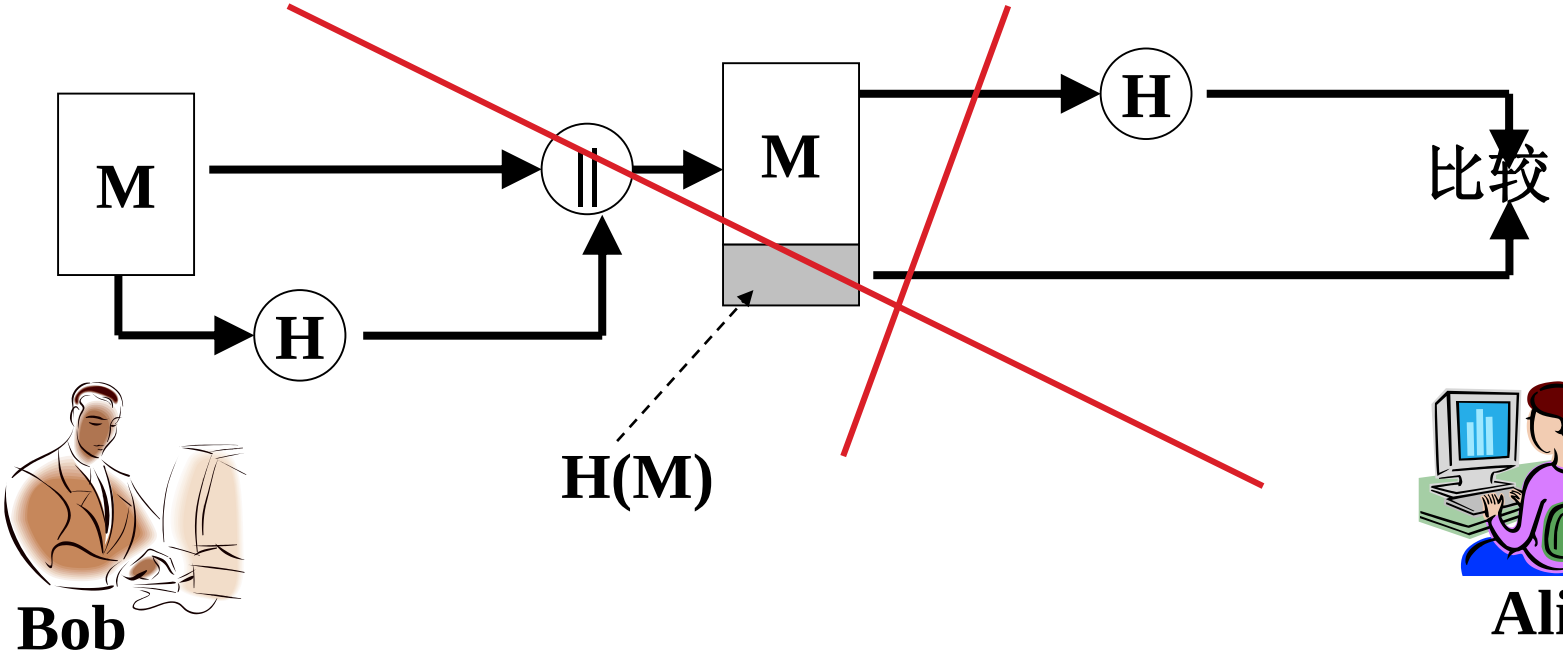
MD5简介

- Merkle1989年提出hash function模型，Ron Rivest于1990年提出MD4，1992年, MD5 (RFC 1321)
- 输入：512bit块
- 输出：128bit
- 2004年前最主要hash算法，在国内外有着广泛的应用，曾一度被认为非常安全。
- 现行美国标准SHA-1以MD5前身MD4为基础。

来自中国的惊艳

- 2004年国际密码学会议（Crypto'2004）山东大学王小云教授做了破译MD5、HAVAL128、MD4和RIPEMD算法的报告。
  - 可以很快找到MD5“碰撞”——两个文件产生相同MD5“指纹”。
  - 意味着：在网络上电子签署一份合同后，还可能找到另一份具有相同签名但内容迥异的合同，这样两份合同的真伪性便无从辨别。
- 宣告固若金汤的世界通行密码标准MD５的堡垒轰然倒塌，引发了密码学界的轩然大波。

哈希函数的基本用法

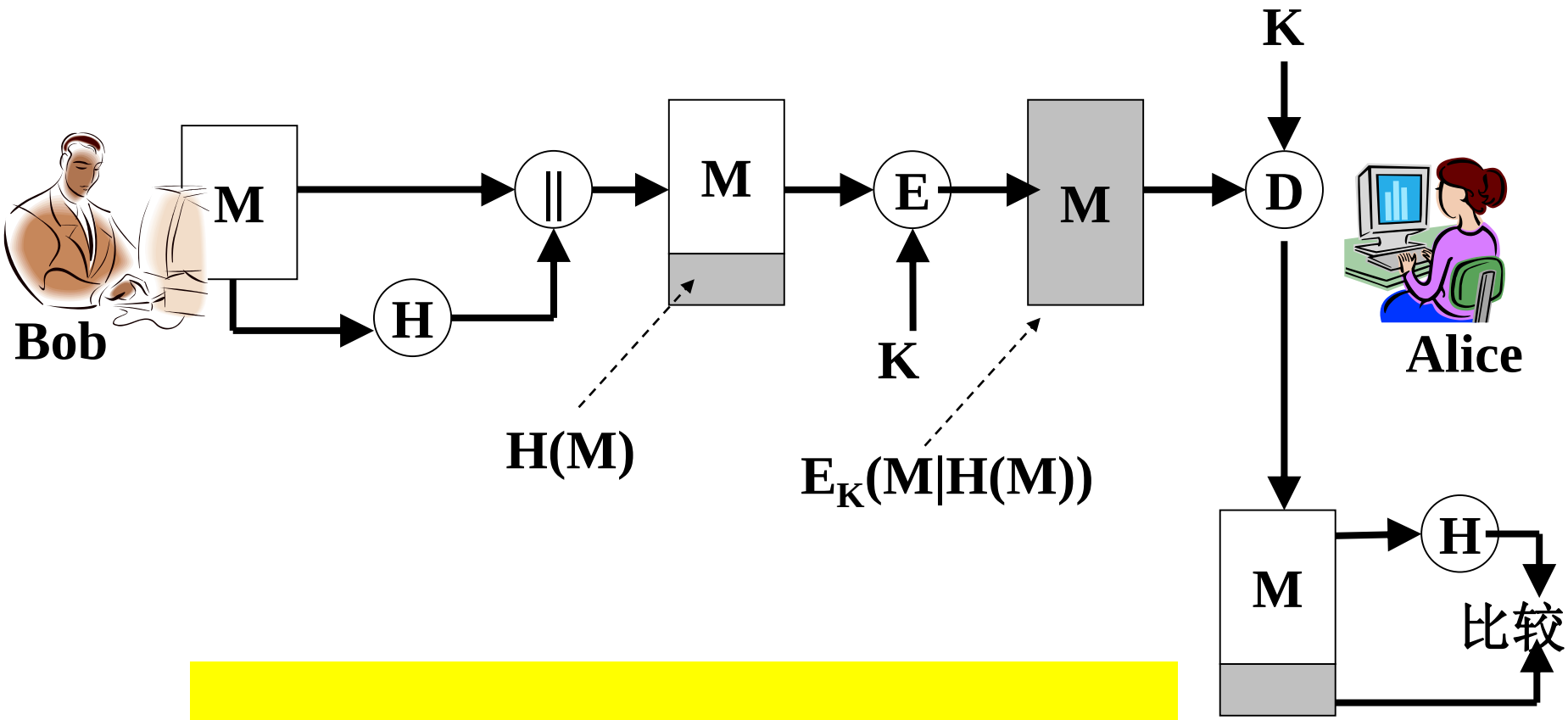


存在问题：没法认证

伪造消息+伪造摘要

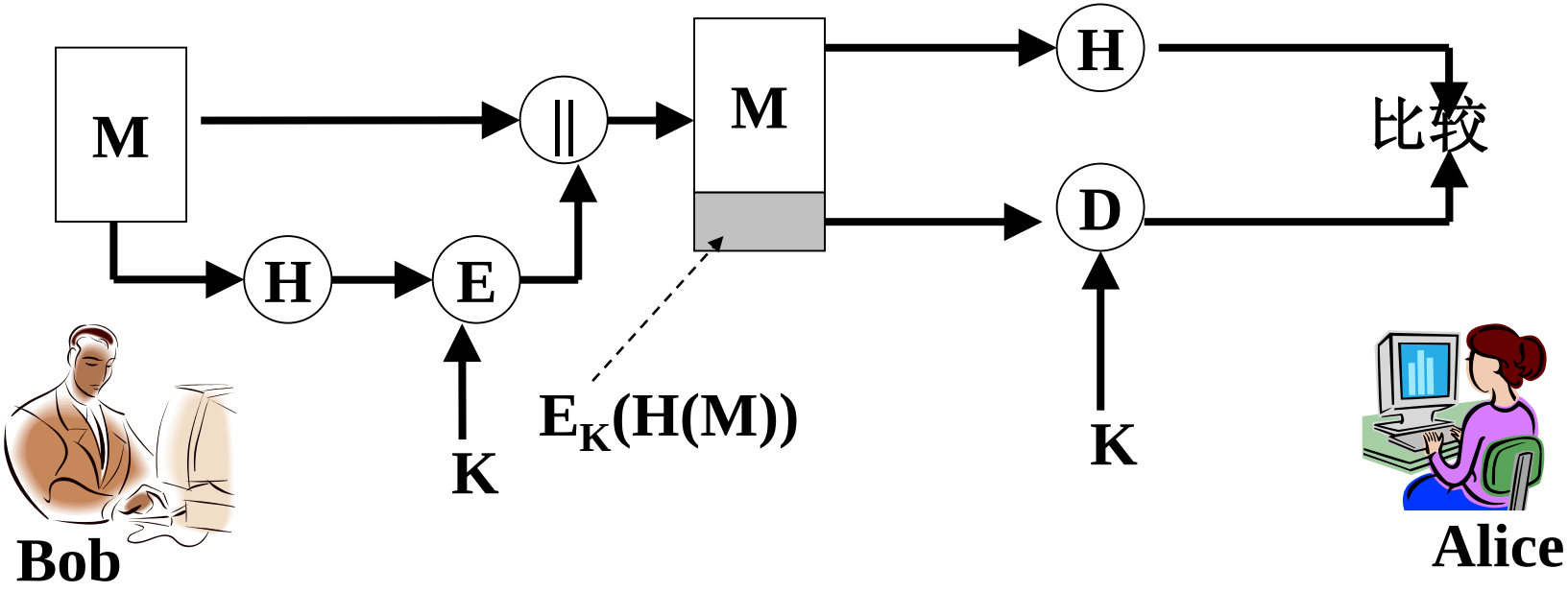
M1+H(M1)

哈希函数的基本用法（a）



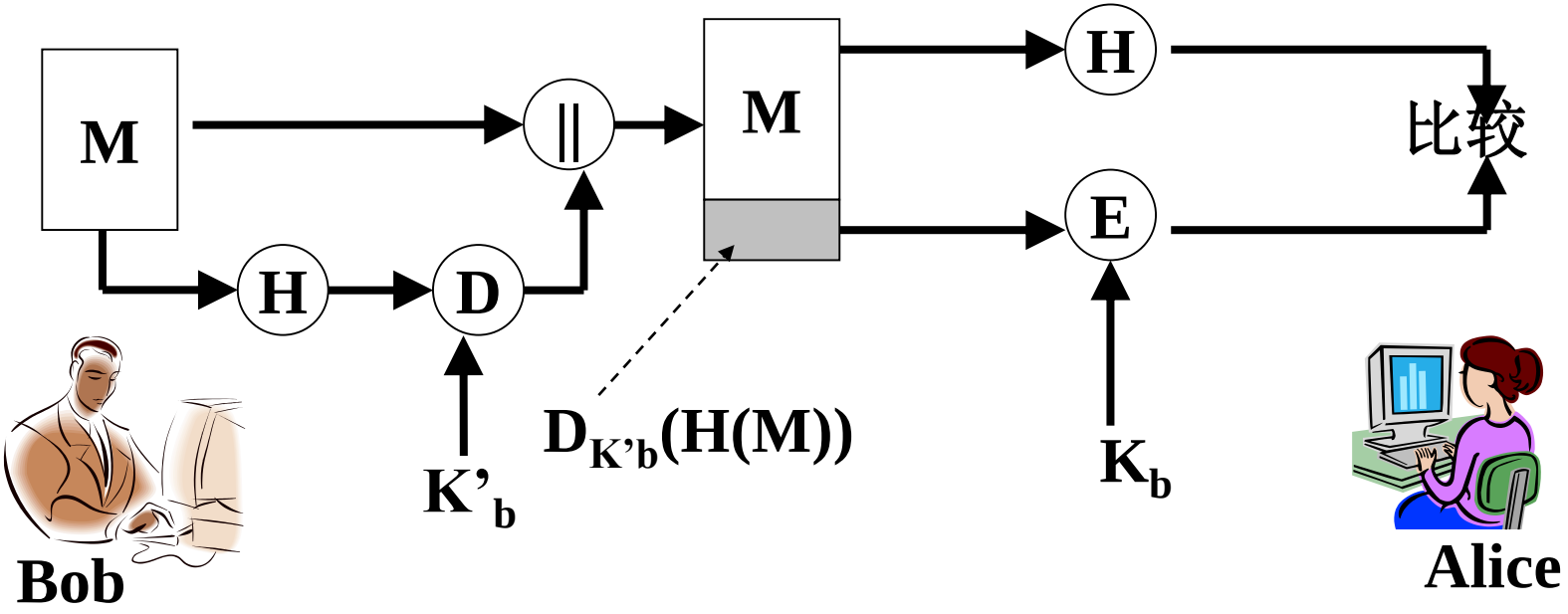
认证+保密

哈希函数的基本用法（b）



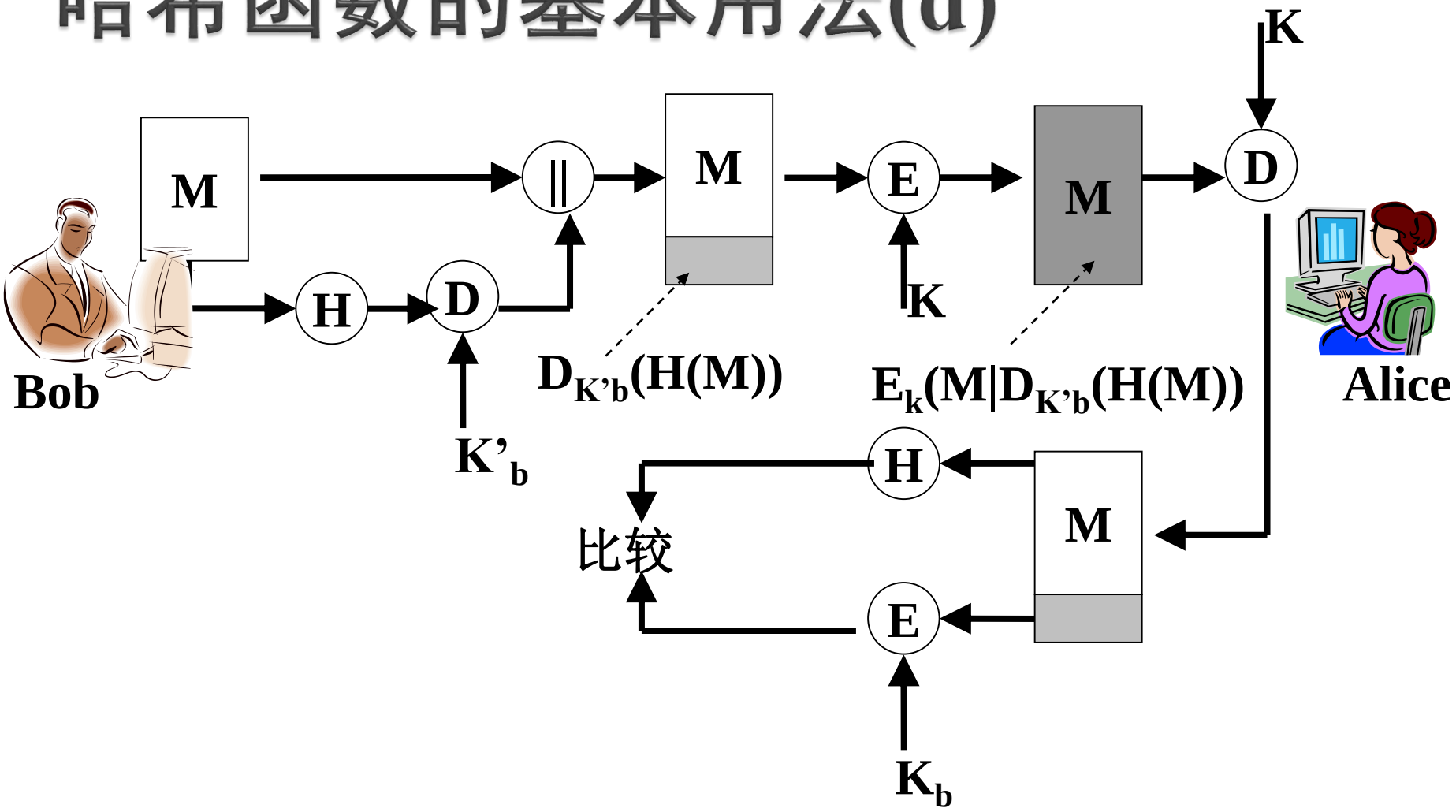
加密hash，仅认证

哈希函数的基本用法（c）



使用公钥密码对hash签名：
提供认证
提供签名

哈希函数的基本用法(d)



完整模型：认证+签名+保密

5.2 数字签名

数字签名需求

- （无签名）消息认证保证完整性/真实性：
  - 保护通信双方数据交换不被第三方侵犯
- 不保证不可否认性，通信双方相互欺骗，如：
  - B伪造消息，声称从A收到的。
  - B收到A发送的消息，A否认发过。
- 例如：通过Email向股票经纪人发出执行某[illegible]损后抵赖发出过命令。

数字签名

数字签名

- 防止源点或终点否认（抵赖）的认证技术
- 传统（笔迹）签名的模拟，传统签名基本特点：
  - 能与被签的文件在物理上不可分割
  - 签名者不能否认自己的签名
  - 签名不能被伪造
  - 容易被验证
- 数字签名是传统签名的数字化，基本要求：
  - 能与所签文件“绑定”
  - 签名者不能否认自己的签名
  - 签名不能被伪造
  - 容易被自动验证

数字签名

- 五元组(M, C, K, S, V)（对应密码算法五元组）
  - M：所有消息组成的有限集
  - C：所有可能的签名组成的有限集
  - K：所有可能的密钥组成的有限集
  - S：签名算法
  - V：验证算法

$$sig_K: M \rightarrow C$$

$$ver_K: M \times C \rightarrow \{真，假\}$$

$$ver_K(\mathrm{m}, c) = \begin{cases} 真, & \mathrm{c} = sig_K(m) \\ 假, & \mathrm{c} \neq sig_K(m) \end{cases}$$

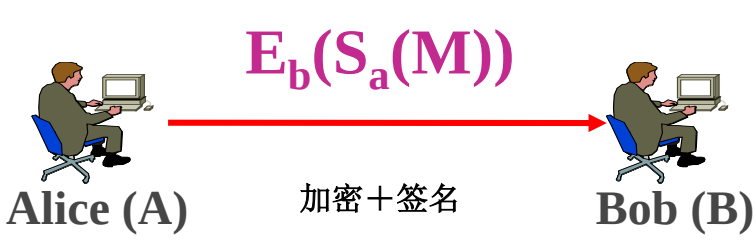
数字签名设计要求

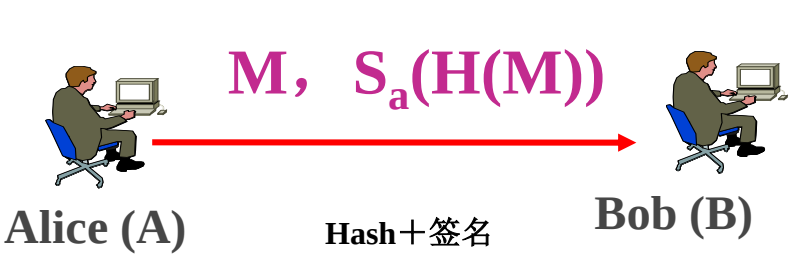
- 签名是被签名信息的相关二进制串；
- 签名必须使用签名者唯一的信息；
- 容易生成数字签名；
- 容易验证数字签名；
- 伪造签名计算上不可行
  - 已有签名伪造新的消息
  - 给定消息伪造数字签名
- 在存储器中保存数字签名副本可行

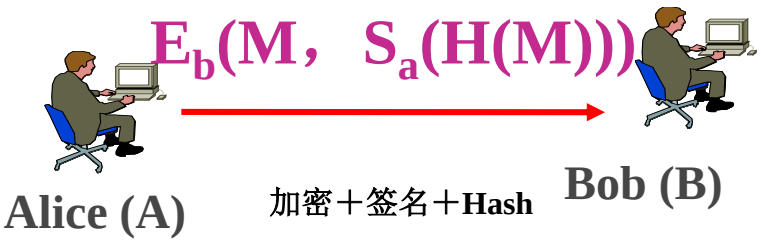
数字签名分类

- 以方式分
  - 直接数字签名
  - 仲裁数字签名
- 以安全性分
  - 无条件安全的数字签名
  - 计算上安全的数字签名
- 以可签名次数分
  - 一次性的数字签名
  - 多次性的数字签名

直接数字签名







直接数字签名缺点

- 签名有效性依赖于发方私钥安全性；
  - 发方私钥丢失或被盗用，攻击者就可以伪造签名。
  - 发送方抵赖：声称私有密钥丢失或被窃，他人伪造签名；
- 改进：
  - 签名包含时间戳，并要求私钥暴露及时报告给授权中心；
- 敌方可伪造早于或等于时间T的时间戳：
  - 时间戳不可信，签名者可自己加时间戳（伪造）

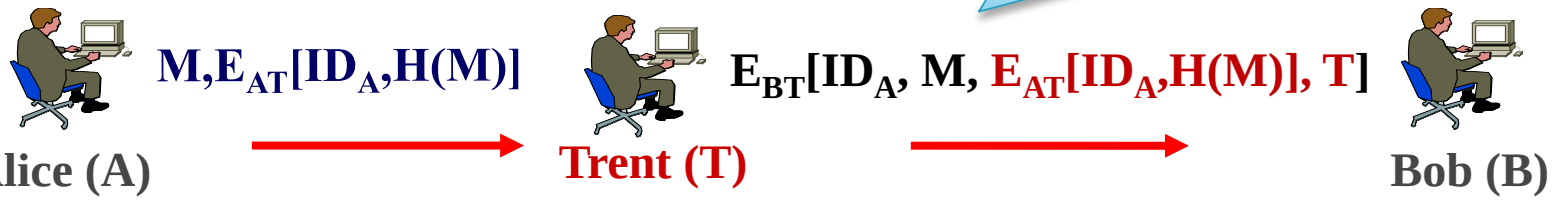
仲裁数字签名

- 引入仲裁者
  - 发送方将签名消息首先送到仲裁者；
  - 仲裁者测试消息及其签名，以检查其来源和内容；
  - 然后将消息加上时间戳，并与仲裁验证通过指示一起发给接收者。
- 仲裁者扮演敏感和关键角色。
  - 所有参与者必须极大地相信这一仲裁机制工作正常。（trusted system）

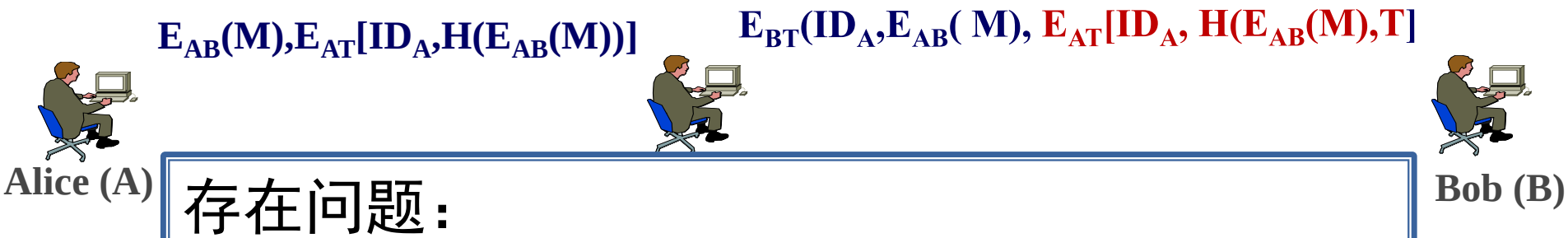
仲裁签名——对称密码

明文传送

发送方没法抵赖
接收方没法伪造



仲裁可见明文，密文传送



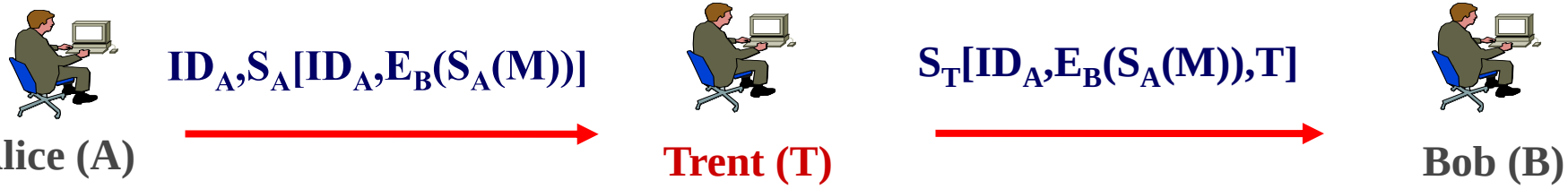
存在问题：

发方与仲裁可结盟来否认一个签名，

或收方与仲裁结盟来伪造一个签名。

使用公开密码算法解决这个问题。

仲裁签名——公钥密码＋密文传送



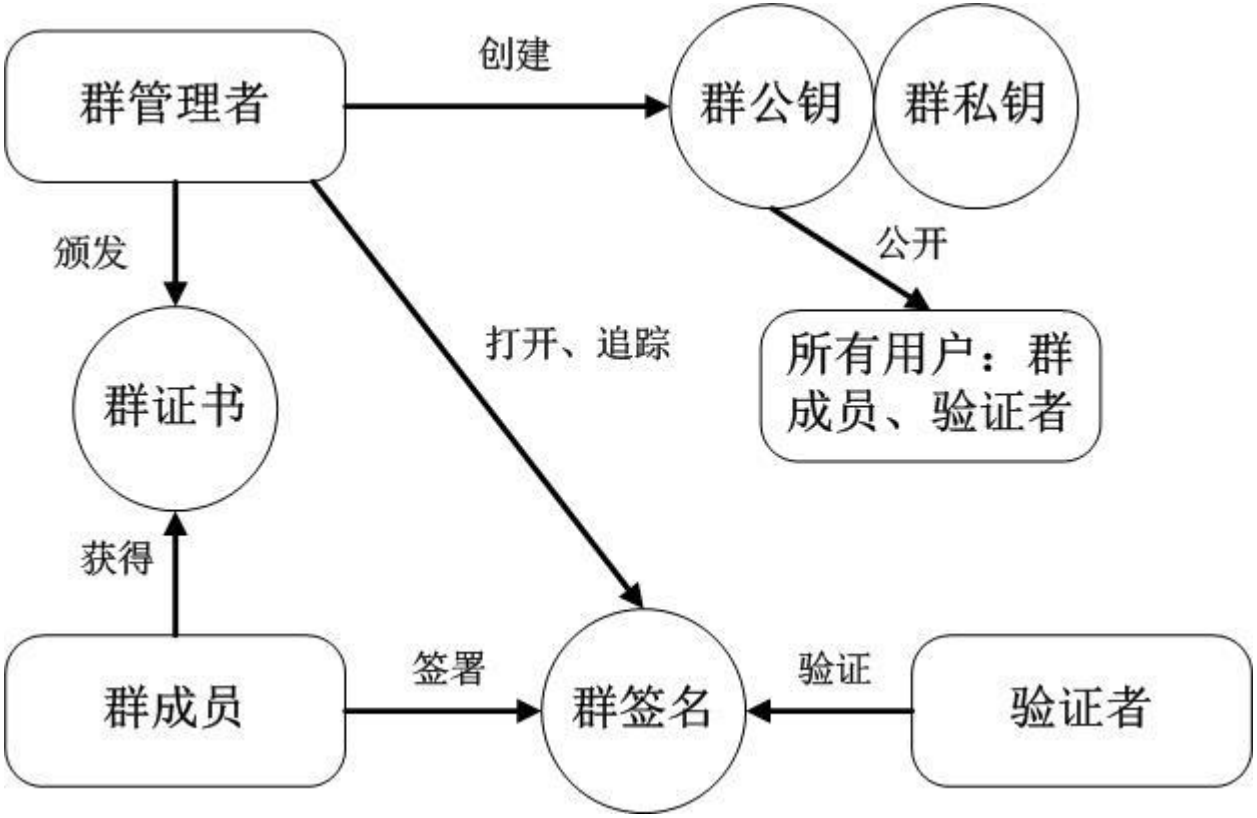
数字签名算法

- 普通数字签名算法
  - RSA
  - ElGamal
  - DSS/DSA
- 群签名算法
- 盲签名算法

群签名方案

- Chaum和van Heyst1991年提出，
- 群中各个成员以群的名义匿名地签发消息，特性：
  - 只有群成员能代表所在的群签名
  - 接收者能验证签名所在的群，但不知道签名者
  - 需要时,可借助于群成员或者可信机构找到签名者
- 应用：投标

群签名过程



盲签名

- 保护消息内容对签名者不可见
- Chaum1983年提出，电子商务领域广泛应用：
  - 电子货币
    - 电子现金须加银行数字签名才能生效，盲签名保护消费者的匿名性；
  - 电子选举
    - 选民提交的选票须盖上选委会的数字签名才合法，盲签名保护选民匿名性。

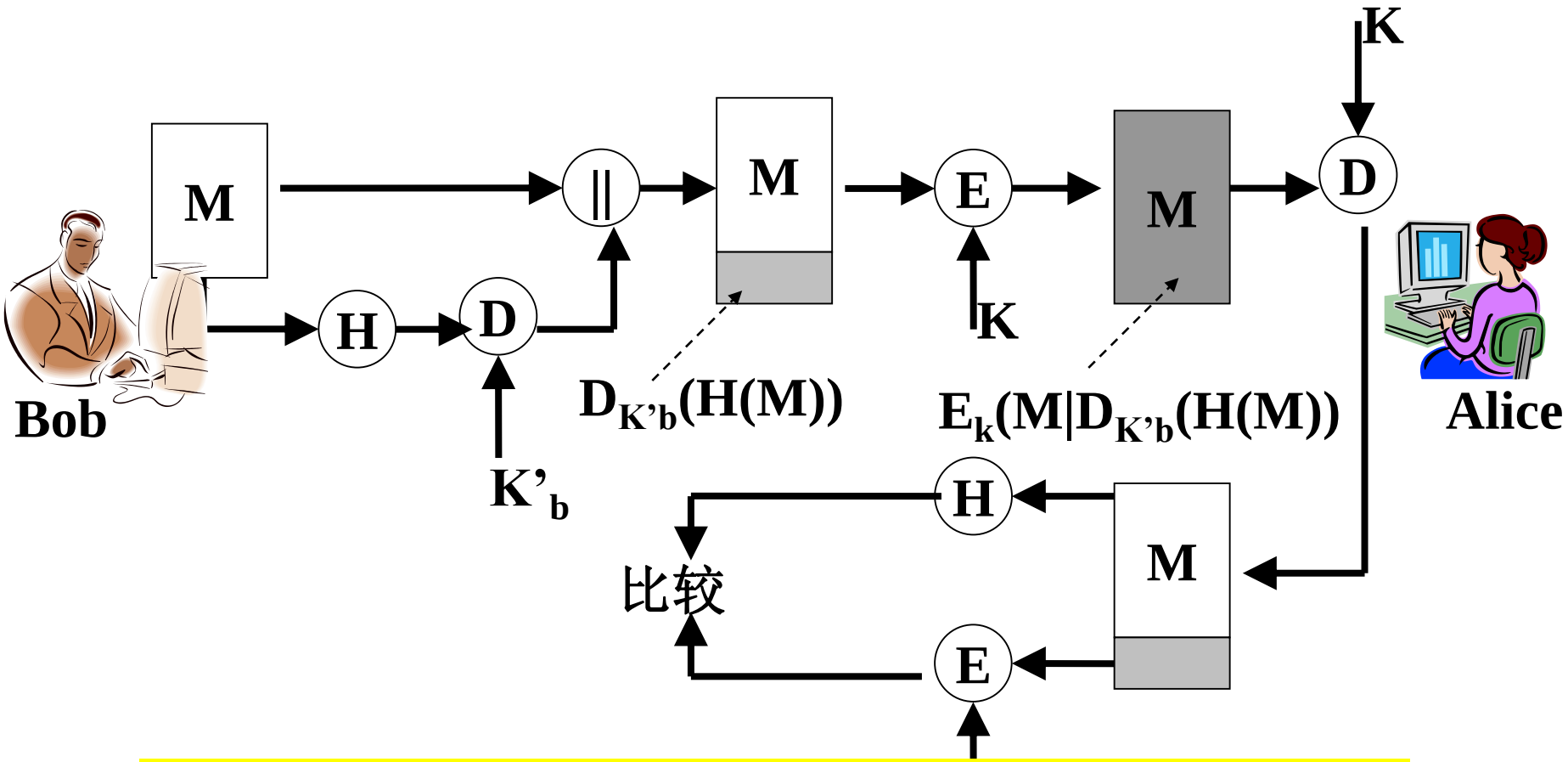
盲签名性质

- 除了满足一般数字签名条件外，还须满足下面两条性质：
  - 签名者不知道其所签名的消息的具体内容。
  - 签名消息不可追踪，即当签名消息被公布后，签名者无法知道这是他哪次的签署的。

盲签名步骤

- 盲化：
  - 消息发送者先将消息盲化
- 签名：
  - 让签名者对盲化的消息进行签名
- 去盲：
  - 消息拥有者对签名除去盲因子，得到签名者关于原消息的签名。

温故而知新——消息认证完整模型



认证+签名+保密

现实中盲签名

- 盲化：将盲签的文件放进信封；
- 签名：信封里放一张复写纸，签名者签信封，签名透过复写纸签到文件上
- 去盲：打开信封

盲RSA签名方案

- Bob从Alice处获得盲签名

$[e,n],\ d$

$t=mr^e \bmod n$

$t^d \bmod n=(mr^e)^d \bmod n$

$=m^d r \bmod n$

$t^d r^{-1}=m^d r \bmod n \times r^{-1}$

$=m^d \bmod n$

消息认证与数字签名的区别

- 消息认证：验证消息真实性及完整性，防范第三者；
- 数字签名：收发双方产生利害冲突时，防止纠纷。