# 第2章 密码体制与技术

# 本章内容

- 2.1 密码学入门
- 2.2 对称密码
- 2.3 非对称密码
- 2.4 数字签名

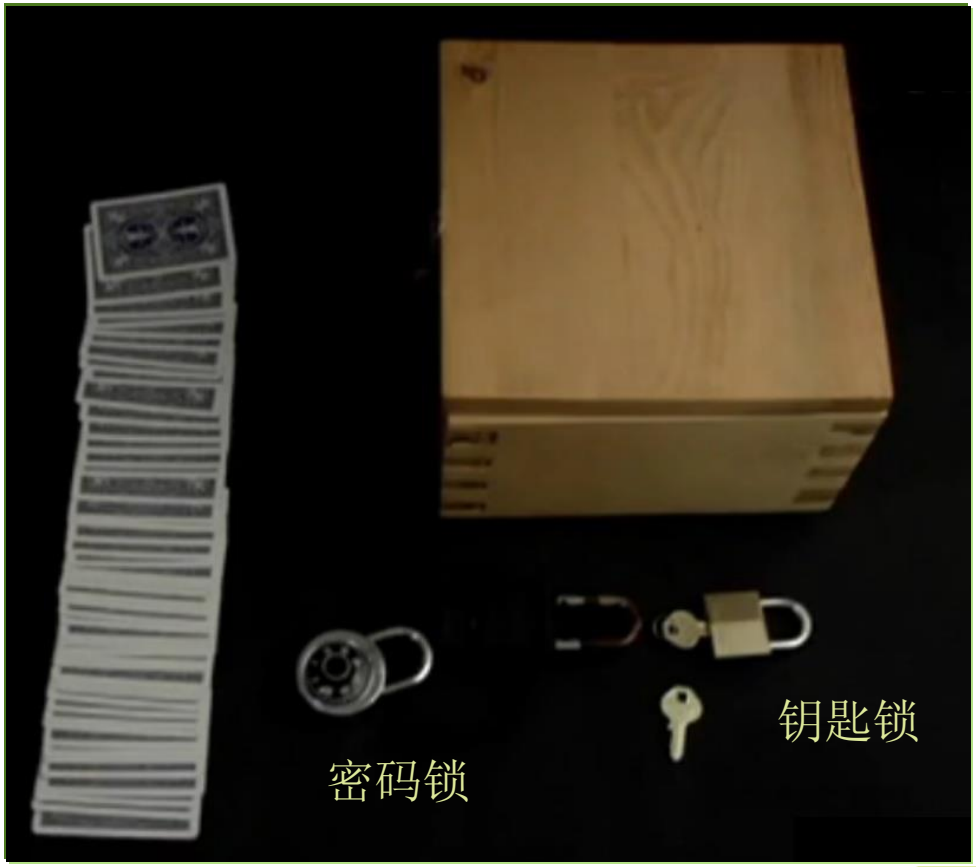
# 2.1　密码学入门

# Alice和Bob的故事

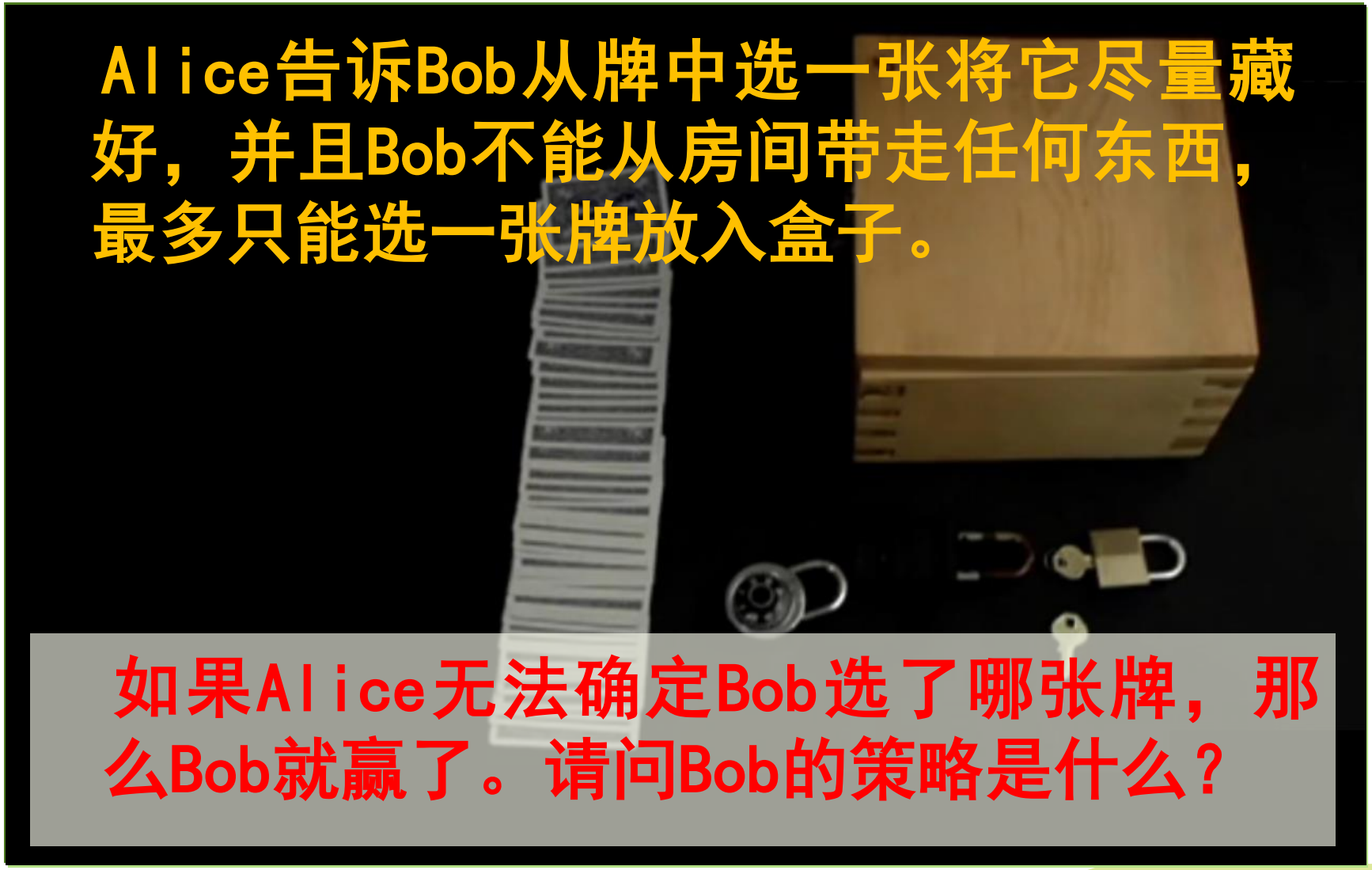
- Alice请Bob进入一个房间

# Alice和Bob的故事

- 房间是空的，只有一些锁、一个空盒子、一副牌。



# Alice和Bob的故事

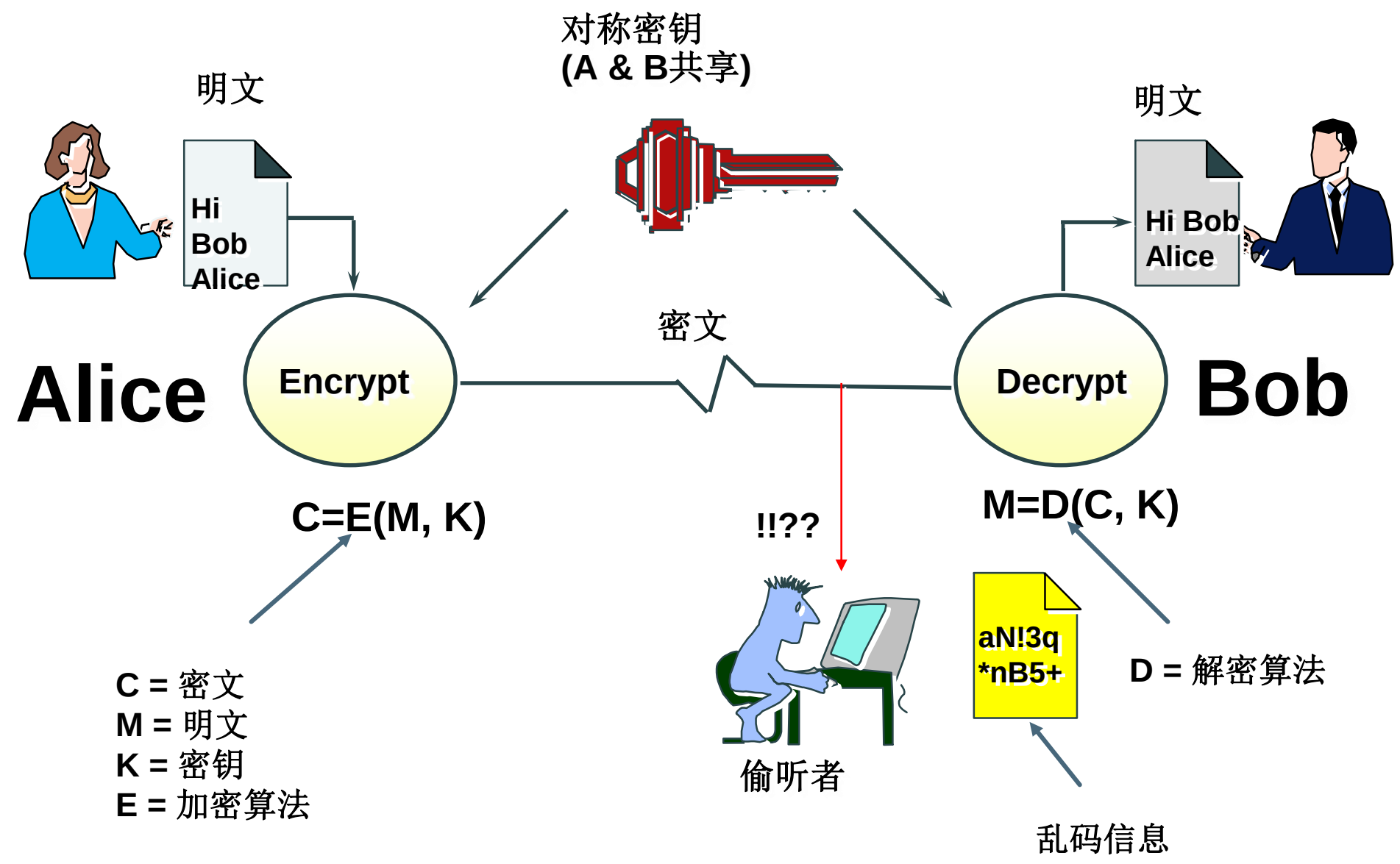


# Alice、Bob 和他们的“朋友们”

表 2-1　剧中人

| 人　名 | 角　色 |
| --- | --- |
| Alice | 所有协议中的第一个参加者 |
| Bob | 所有协议中的第二个参加者 |
| Carol | 三、四方协议中的参加者 |
| Dave | 四方协议中的参加者 |
| Eve | 窃听者 |
| Mallory | 恶意的主动攻击者 |
| Trent | 值得信赖的仲裁者 |
| Walter | 监察人：在某些协议中保护 Alice 和 Bob |
| Peggy | 证明人 |
| Victor | 验证者 |

# 2.2　对称密码


对称密钥
(A & B共享)
明文
Hi
Bob
Alice
明文
Hi Bob
Alice
Alice
Encrypt
密文
Decrypt
Bob
C=E(M, K)
!!??
M=D(C, K)
aN!3q
*nB5+
C = 密文
M = 明文
K = 密钥
E = 加密算法
偷听者
D = 解密算法
乱码信息

# 你能破译吗？

- 下面是一段密文，请问明文和密钥分别是什么？

AVCIC GN RJ SGLGA AJ AVC VCGEVAN
THE E T T THE HE HT
CYCIK LPR MPR PAAPGR XH ICLPGRGRE
E E TT E
JR AVC SCYCS
THE E E

# 你能破译吗？

- 下面是一段密文，请问明文和密钥分别是什么？

AVCIC GN RJ SGLGA AJ AVC VCGEVAN

THERE IS NO LIMIT TO THE HEIGHTS

CYCIK LPR MPR PAAPGR XH ICLPGRGRE

EVERY MAN CAN ATTAIN BY REMAINING

JR AVC SCYCS

ON THE LEVEL

ABCDEFGHIJKLMNOPQRSTUVWXYZ

# 对称密码体制的优点

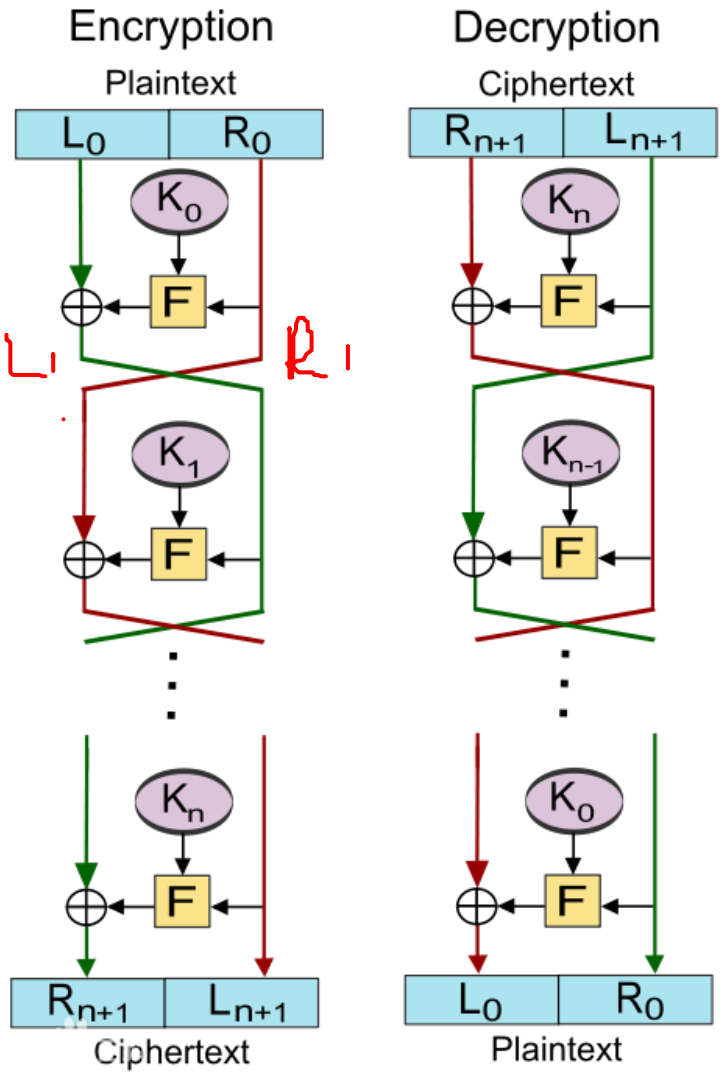
- 经典对称加密算法
  - DES，3DES
  - AES
  - Blowfish
  - RC5
  - IDEA，TDEA

- 对称密码体制的优点：
  - 算法简单、速度快、适合加密大量数据

# DES加密算法

- **发明人：**
  - IBM公司 W.Tuchman和C.Meyer
- **思想起源：**
  - 参照二战德国的恩尼格码机，混淆和扩散
- **密钥长度：**
  - 64位，但每个第8为都用作奇偶校验，所以对于使用者而言，密钥长度是56位。
- **基本操作：**
  - 采用Feistel技术，16个循环，异或、置乱、替代、移位四种基本运算。

# DES加密算法——Feistel结构

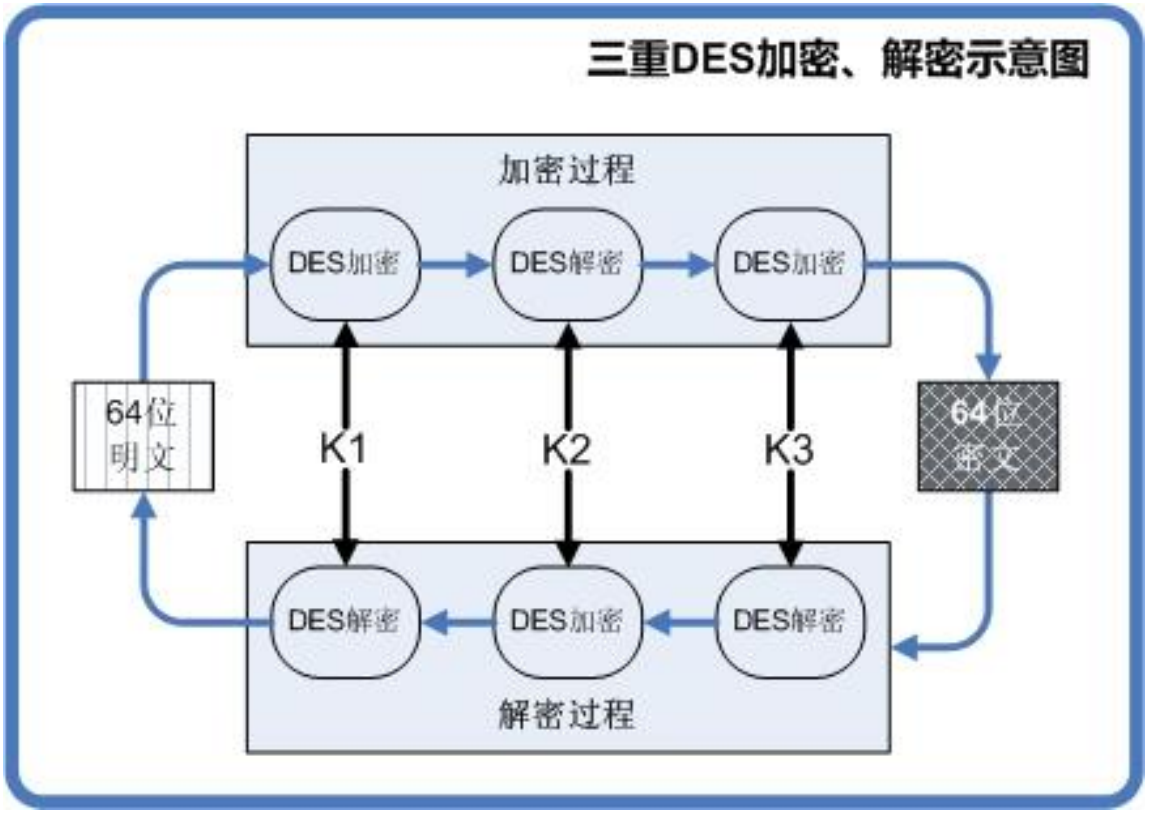



❖思考：

Feistel结构的优点和缺点是什么？

# 3DES加密算法

- 目的：解决DES密钥短的问题。




- 思考：

为什么是“加密-解密-加密”模式，而不是“加密-加密-加密”模式？

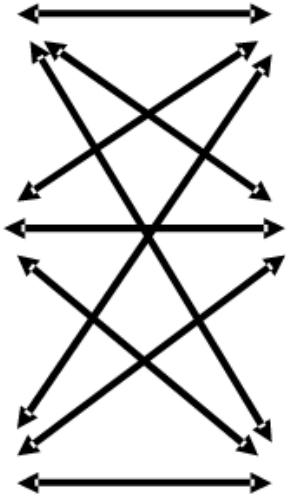
# AES加密算法

- 高级加密标准，也叫Rijndael算法
- 由两位比利时密码学家发明，参与了NIST（美国标准和技术委员会）1997年组织的公开密码学竞赛，最终以优异的技术特性胜出成为加密标准。
- 分组长度和密钥长度可变

# AES加密算法

表 1. 分组长度和密钥长度的不同取值

| 分组长度（bit） | 密钥长度（bit） |
| --- | --- |
| 128 | 128 |
| 192 | 192 |
| 256 | 256 |



# 对称密码体制的弱点

## 密钥管理

- 如何安全的共享秘密密钥，不可能与你未曾谋面的人通信
- 每对通信者间都需要一个不同的密钥。N个人通信需要$C_N^2$ ＝n(n-1)/2密钥。当用户量增大时密钥空间急剧增大。如：

n= 100 时C( 100,2)= 4,995

n= 5000 时C( 5000,2)= 12,497,500

## 没有解决不可抵赖问题

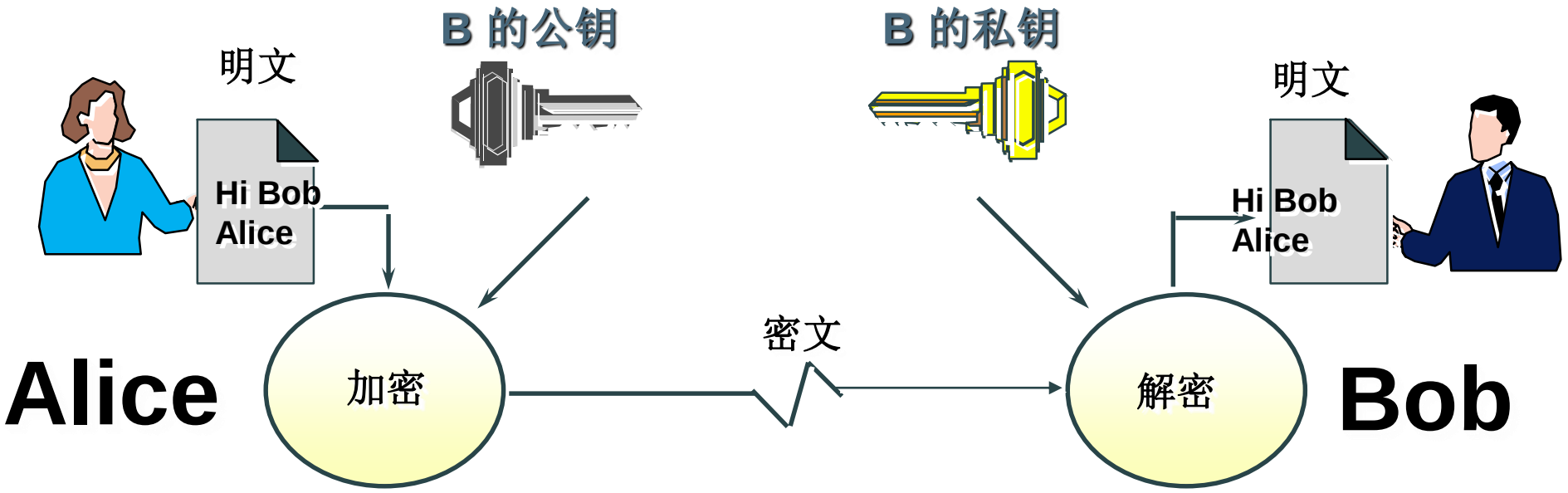
- 文档不能被签名
- 通信双方都可以否认发送或接收过的信息

# 2.3 非对称密码

# 起源

- 非对称密码又称为公钥密码和双钥密码
- Diffie和Hellman于1976年在《密码学的新方向》中首次提出了公钥密码的观点，标志着公钥密码学研究的开始
- W. Diffie and M. E. Hellman, New direction in cryptography, IEEE Trans. on Information Theory, 1976, IT-22.(6), pp.644-654.
- 1977年由Rviest，Shmair，Adlmena提出了第一个比较完善的公钥密码算法，即RSA算法

# 公钥密码的加密过程




- A 发送机密信息给 B, 知道只有 B 可以解密
- A 用 B 的公钥加密 (公开)
- B 使用自己的私钥解密 (保密)

# 公钥密码的原理

## ❖公钥密码的理论基础：单向陷门函数

单向函数：已知x, 计算y使得y=f(x)容易；

已知y, 计算x使得y=f(x)是难解的。

陷门单向函数：t是与f有关的一个参数

已知x, 计算y使得y=f(x)容易；

如果不知道t，已知y, 计算x使得y=f(x)是难的，

但知道t时，已知y, 计算x使得y=f(x)是容易的。

参数t称为陷门。

# 公钥密码的原理

- 设计公钥密码可以转换为寻找单向陷门函数。目前，主要基于如下的数学上的困难问题来设计单向函数和公钥密码体制：
- （1）大整数分解问题

若已知两个大素数p、q，求n=pXq仅需一次乘法，但已知n求p、q则是一道难题。

- （2）有限域上的离散对数问题

对于等式 $y = g^x \bmod p$，给定$g$、$x$和$p$，计算$y$是容易的。

反过来，若已知$y$、$g$和$p$，当$p$是大素数时，找到一个$x$，使 $y = g^x \bmod p$ 成立是困难的。

- （3）椭圆曲线上的离散对数问题

# 公钥密码特点

- 公钥密码算法是基于数学函数而不是基于替代和置换
- 公钥密码是非对称的，它使用两个独立的密钥，即公钥和私钥，任何一个都可以用来加密，另一个用来解密
- 公钥可以被任何人知道，用于加密消息以及验证签名，私钥仅自己知道，用于解密消息和签名

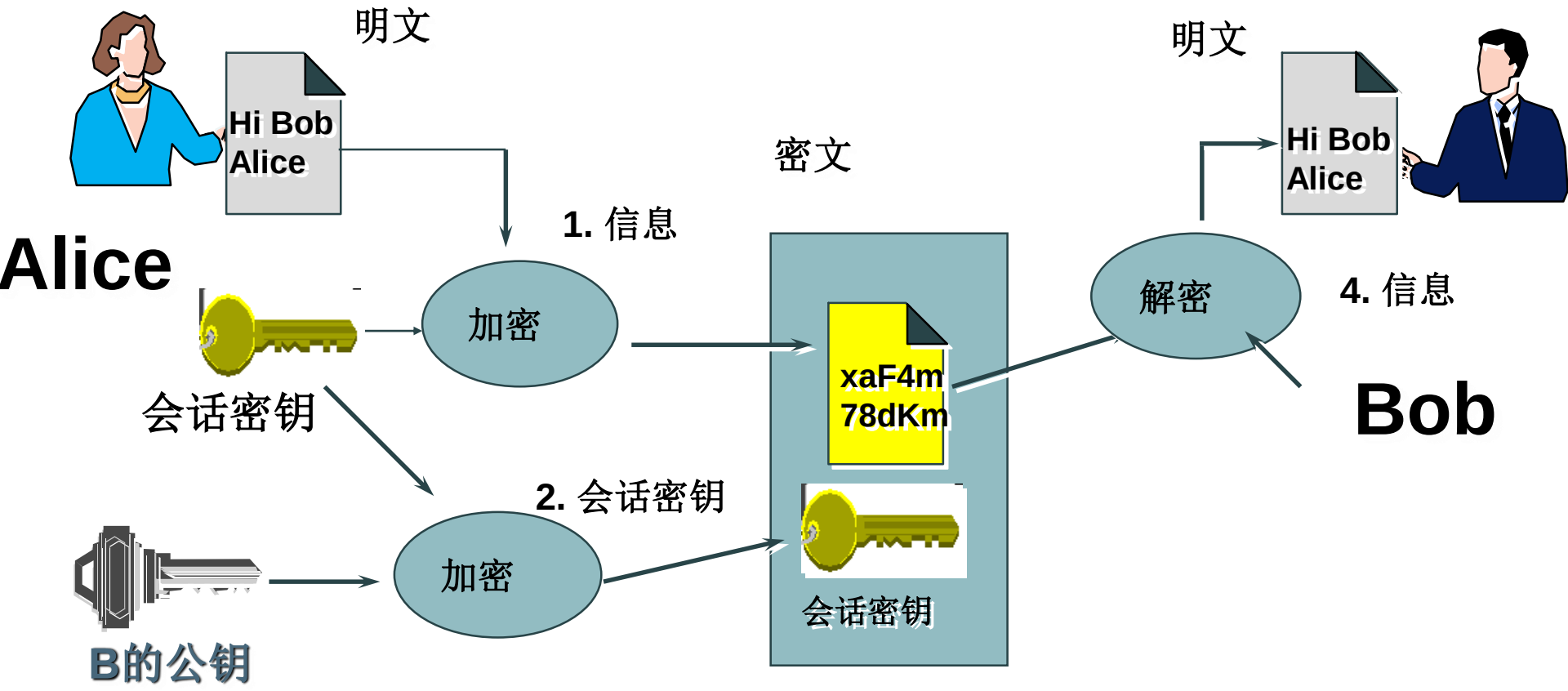
# 常用的公钥密码算法

- RSA
  - Ron Rivest, Adi Shamir和Len Adleman于1977年研制并且1978年首次发表
  - 可以用私钥加密和公钥加密
- DSA
  - 最初由NIST于1991年发布
  - 只能使用私钥加密，通常用作数字签名
- Diffie-Hellman算法
  - 只能用来进行对称密钥交换

# 非对称密码的问题

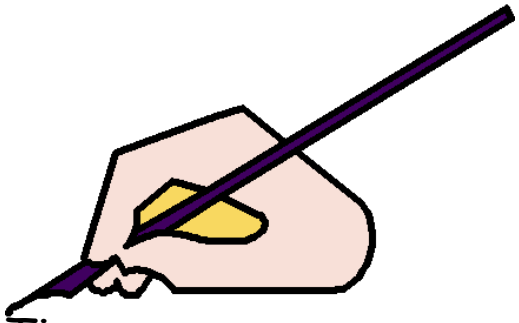
- **公钥加解密对速度敏感**
  - 大数幂运算，因此非常慢
  - 软件，公钥算法比对称密钥算法慢 100 多倍。（硬件可能慢 1000倍 ）
- **公钥加密长信息无法接受的慢，而对称密钥算法非常快**
- **结合公钥算法和对称密钥算法，使用对称密钥与公开密钥的优点**
  - 对称密钥快速而强健
  - 公开密钥易于密钥交换

# 组合对称密码和非对称密码




- 产生一个一次性，对称密钥——会话密钥
- 用会话密钥加密信息
- 最后用接收者的公钥加密会话密钥——因为它很短

# 2.4　数字签名



Digital Signature, Date, Time

# 数字签名技术

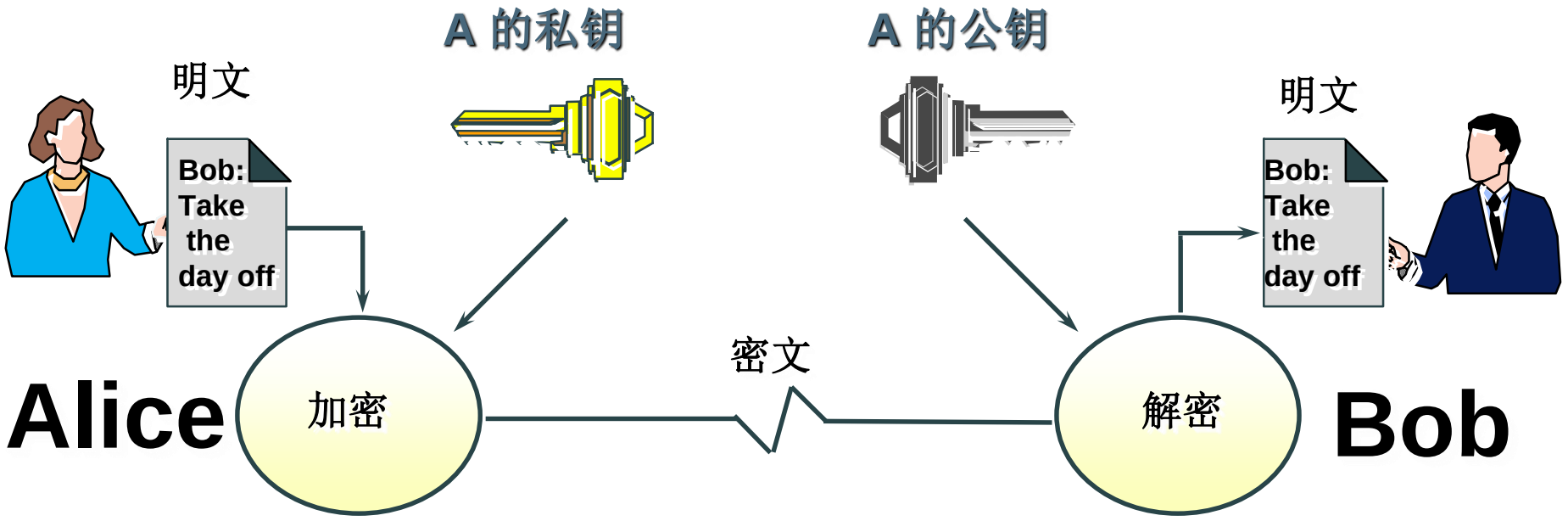
## 数字签名的需求

- Alice 需要一个方法签名一个信息，必须确认是从她发出，因此需要将她的身份和信息绑定在一起。
- 我们用传统的方法将Alice的普通签名数字化后附加在文档的后面
- 但是这个 *数字化* 的签名 ...
  - 它不能避免通过附加在其他文档中被伪造,
  - 无法防止对机密文档（比如支票）的篡改

# 数字签名的实现

- 需要一个数字码唯一标识一个人或实体
  - 身份证号码? No, 不保密
  - 私钥? Yes!
- 公钥与私钥是一对镜像
  - 用其中一个加密, 用另一个解密
- 解决方案：用发送者的私钥加密信息，然后用公钥解密
  - 如果能够解开，说明发送者加密并发送了本信息
  - 除非发送者的私钥不再保密

# 数字签名原理—公钥鉴别




- Alice 用她的私钥加密整个信息
- 所有人都可以解密这个信息
- 因此，Bob 可以确信这个信息是由 Alice 产生的 — 因为只有她的公钥可以解开该信息，而只有Alice 有对应的私钥
- 通过公钥鉴别，可以鉴别签名的真实性。

# 公钥鉴别的问题

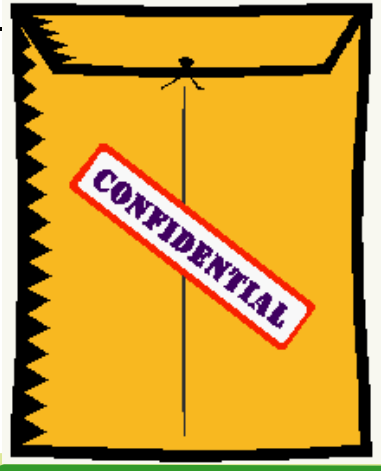
- 问题：签名太长
- 解决方法：签名一个短的信息—数字摘要
- 数字摘要（Message Digest）
  - 一个函数，输入一个任意长度的信息，而输出一个短的固定长度的编码
  - 一般 16 到 20 字节长
  - 对于输入信息 MD 是唯一
  - 无法找到具有相同 MD 的两个信息
  - 对于信息的任何修改，MD将改变

# 数字摘要技术

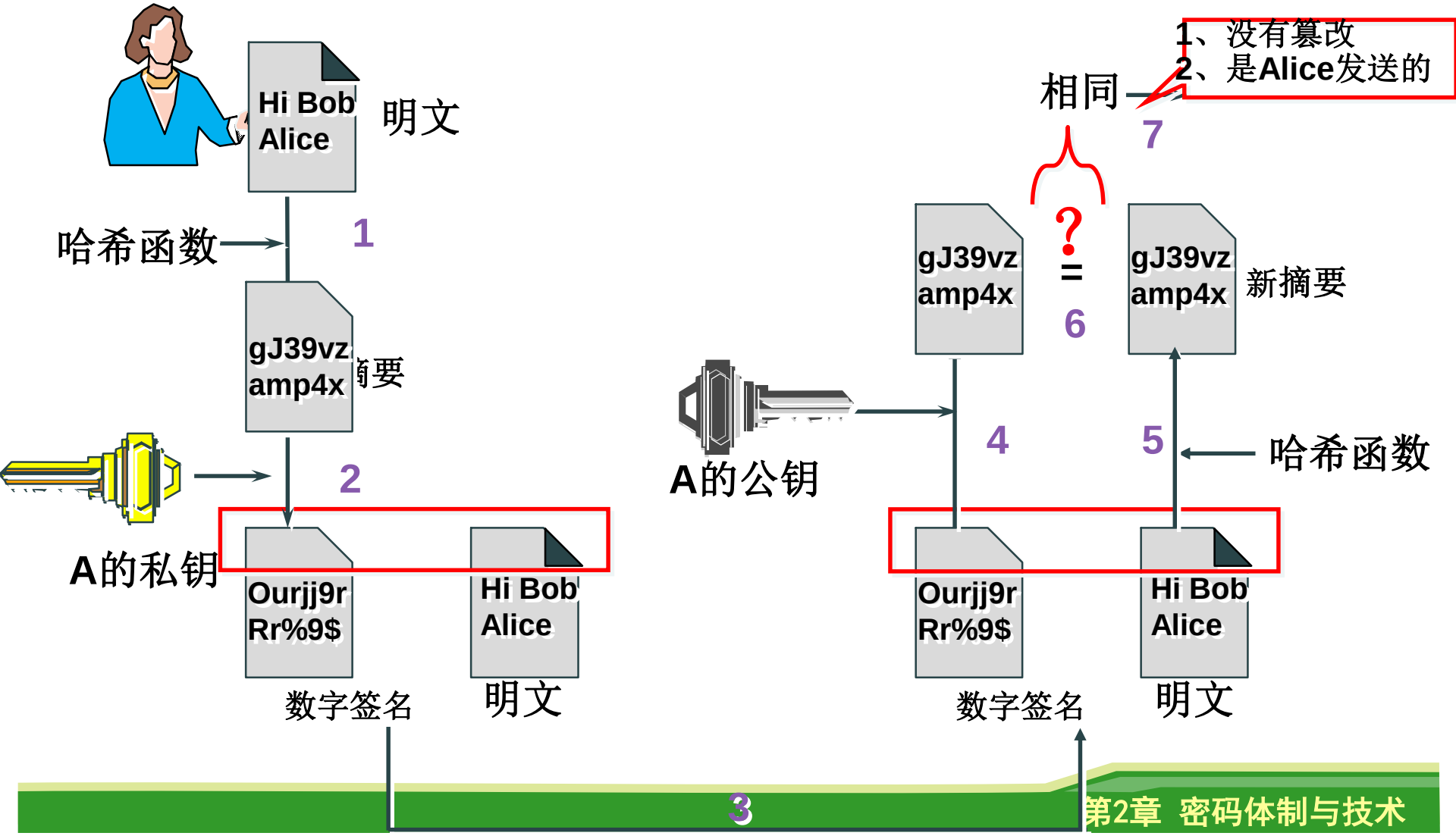
- 散列函数与指纹相象
  - 比原物（本人）信息量小
  - 与本人一一对应
  - 无法找到相同指纹的两个人
  - 知道了指纹，无法重建一个人
- 最常用的散列函数
  - RSA公司的 MD4 和 MD5 （128 位即16 字节）
  - NIST 的安全散列算法 SHA（160 位即20字节）

# 摘要算法 – 数据的完整性

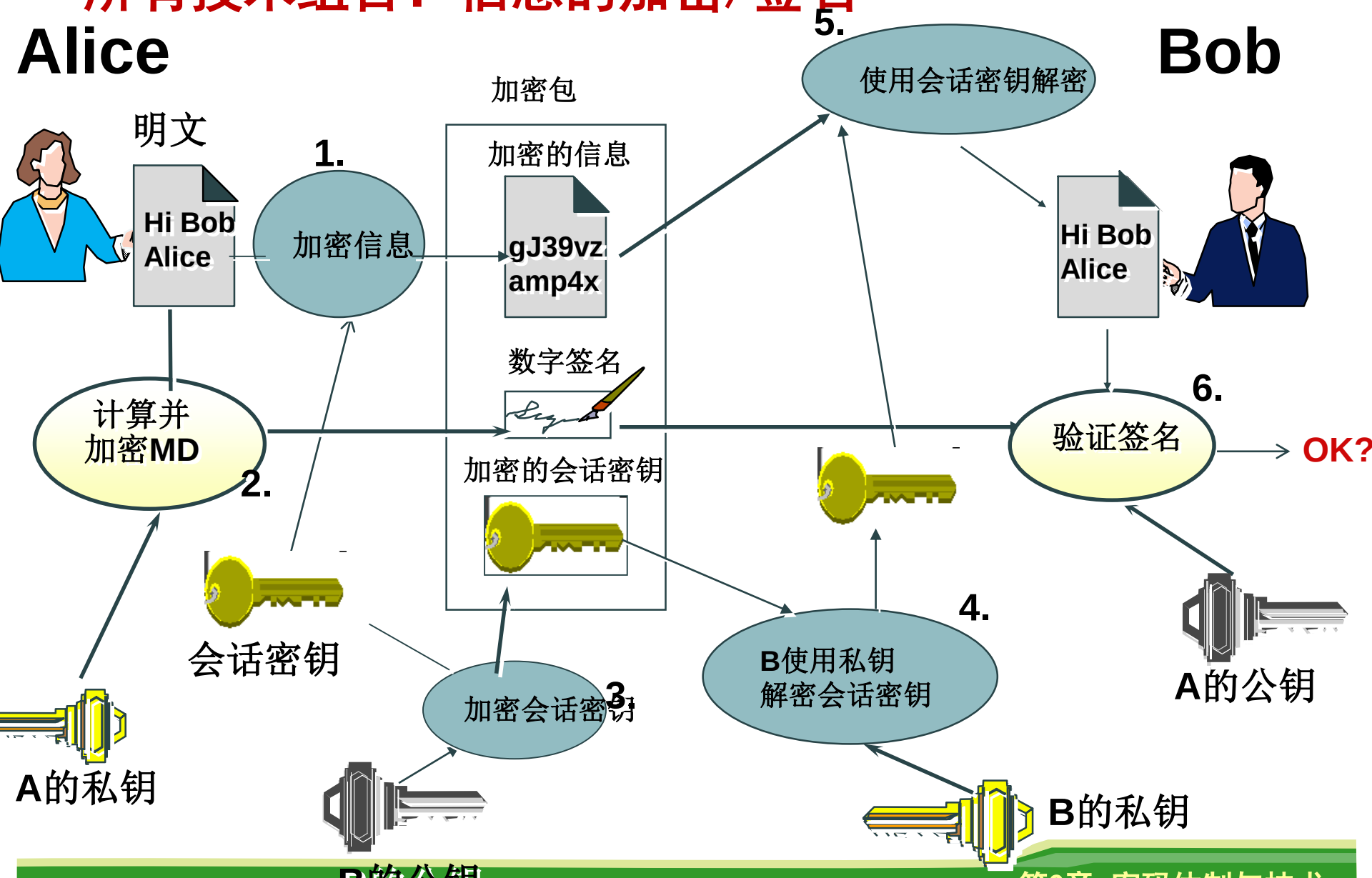
| 输入 | 哈希结果（使用 MD5） |
| --- | --- |
| Could you please transfer $100 from my checking account to the account of Mr.Smith? | D55f1123532d43a16a08557236615502 |
| Could you please transfer $1000 from my checking account to the account of Mr.Smith? | 67b7ba62cae668d8a47bbdf5128a1055 |



# 结合数字摘要的数字签名



# 所有技术组合：信息的加密/签名



河海大学


# Thank You!