第二章 密码编码工具

- 2.1 对称加密
- 2.2 消息认证和散列函数
- 2.3 公钥加密
- 2.4 数字签名和密钥管理
- 2.5 随机数和伪随机数
- 2.6 实际应用：存储数据的加密

2.1 对称加密

- 对称加密也称“传统加密”或“单密钥加密”，是 20 世纪 70 年代后期公钥密码产生之前唯一的加密技术，至今仍是使用最广泛的加密算法之一。

- 其基本成分包括
  - 明文（原始可理解的消息和数据）
  - 加密算法（对明文进行代换和变换）
  - 秘密密钥（加密算法的输入，特定代换和变换依赖于它）
  - 密文（算法输出的随机杂乱数据，依赖于明文和密钥）
  - 解密算法（加密算法的逆运算，输入密文和密钥可恢复明文）。

2.1 对称加密

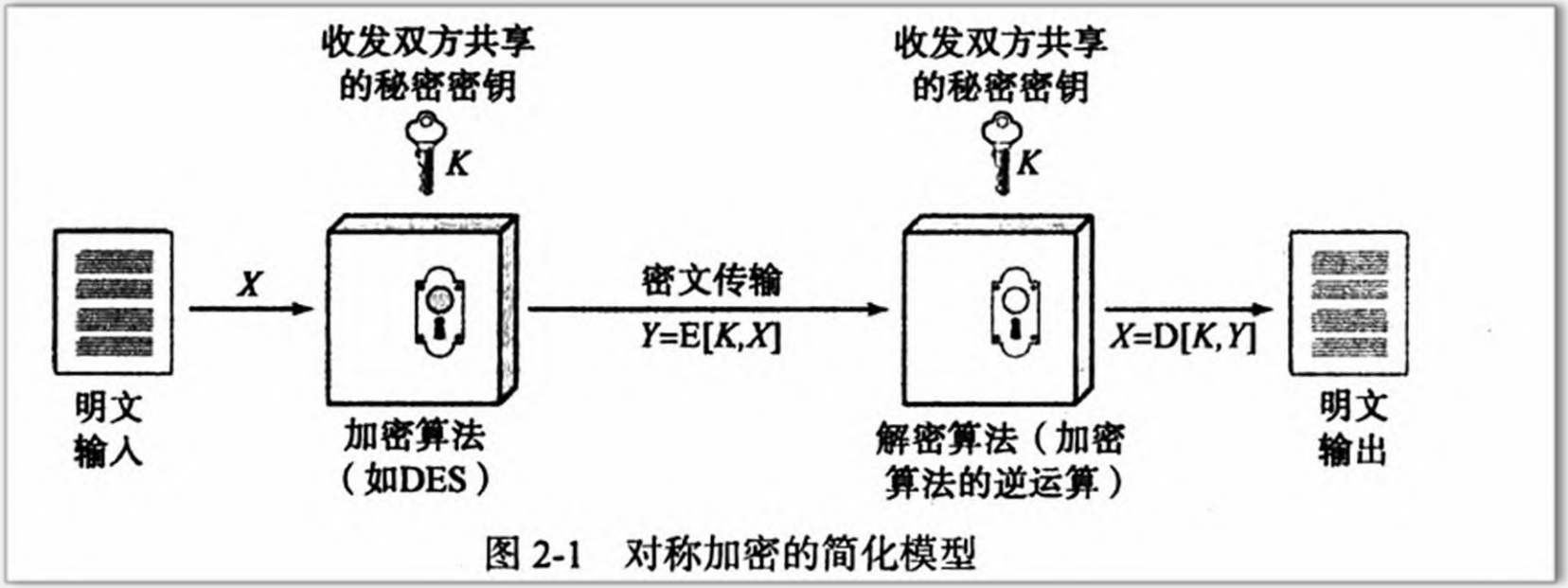


图 2-1　对称加密的简化模型

在对称加密中，加密和解密使用相同的密钥。就好像是一把特殊的“锁”和对应的“钥匙”，发送方用这把“钥匙”（密钥）将明文信息加密成密文，就像是把信息锁进了一个加密的箱子。

2.1 对称加密

- 对称加密的安全使用需满足两个要求:
  - 一是加密算法足够强，敌手即使知道算法和部分密文甚至明文 - 密文对样本也不能破译密文或计算出密钥
  - 二是发送者和接收者要在安全形式下获得并保证密钥安全，否则密钥泄露会危及所有使用该密钥加密的通信内容。

- 攻击对称加密体制的一般方法有:
  - 密码分析学：利用算法性质和明文特征等推导密钥
  - 蛮力攻击：尝试所有可能密钥，平均需尝试一半，明文已知时可能尝试所有密钥，对于压缩的消息或一般数据类型识别较难，需相关明文知识和自动识别方法辅助

2.1 对称分组加密

- 分组密码是使用最广泛的对称加密算法，其原理是将定长的明文转换成与明文等长的密文，并将较长明文划分为一系列定长的块。

- 最重要的对称算法，如数据加密标准（DES）、三重DES（3DES）和高级加密标准（AES），这些算法都使用分组密码。

2.1 对称分组加密

- 数据加密标准（DES），它是使用最广泛的加密体制，1977 年被美国国家标准局（现国家标准和技术研究所，NIST）采纳为联邦信息处理标准 FIPS PUB 46。
- DES采用 64 位长度的明文分组和 56 位长度的密钥，产生 64 位长度的密文分组。
- DES（数据加密标准）的强度：就算法本身，多年来人们尝试发现其弱点，但至今无致命弱点报道；而对于 56 位密钥的使用，考虑到现在的商用处理器速度，该密钥长度是严重不足的。

2.1 对称分组加密

密钥长度（位）	加密算法	可选密钥个数	按 10^9 次解密 /μs 计算所需时间	按 10^{13} 次解密 /μs 计算所需时间
56	DES	$2^{56}\approx7.2\times10^{16}$	$2^{55}\mu s=1.125$ 年	1 小时
128	AES	$21^{28}\approx3.4\times10^{38}$	$2^{127}\mu s=5.3\times10^{21}$ 年	5.3×10^{17} 年
168	3DES	$21^{68}\approx3.7\times10^{50}$	$2^{167}\mu s=5.8\times10^{33}$ 年	5.8×10^{29} 年
192	AES	$21^{92}\approx6.3\times10^{57}$	$2^{191}\mu s=9.8\times10^{40}$ 年	9.8×10^{36} 年
256	AES	$22^{56}\approx1.2\times10^{77}$	$2^{255}\mu s=1.8\times10^{60}$ 年	1.8×10^{56} 年

由该表可见，一台 PC 破解一个密钥长度为 56 位的 DES 密码需要 10 小时；如果由多个 PC 并发运行，那么所需时间会大大缩短；而当前的超级计算机在 1 小时内就应该能够完成这个运算。128 位或更多位密钥能保证算法不会被简单蛮力破解，即使设法将破解机的速度提高 10^{12} 倍，仍然需要 100000 年的时间来破解密码。

2.1 三重DES算法

- 三重 DES（3DES）算法，通过重复基本的 DES 算法三次，采用两个或三个不同的密钥，密钥长度为 112 位或 168 位。
- 3DES 于 1985 年在 ANSI 标准 X9.17 中首次标准化并应用于金融领域，1999 年被合并为数据加密标准（DES）的一部分。
- 其优点是密钥长度长，能克服 DES 的蛮力攻击问题，对密码分析攻击有强免疫力。
- 缺点是软件实现速度慢，要求三倍于 DES 的计算量，且分组长度与 DES 相同均为 64 位。

2.1 高级加密标准

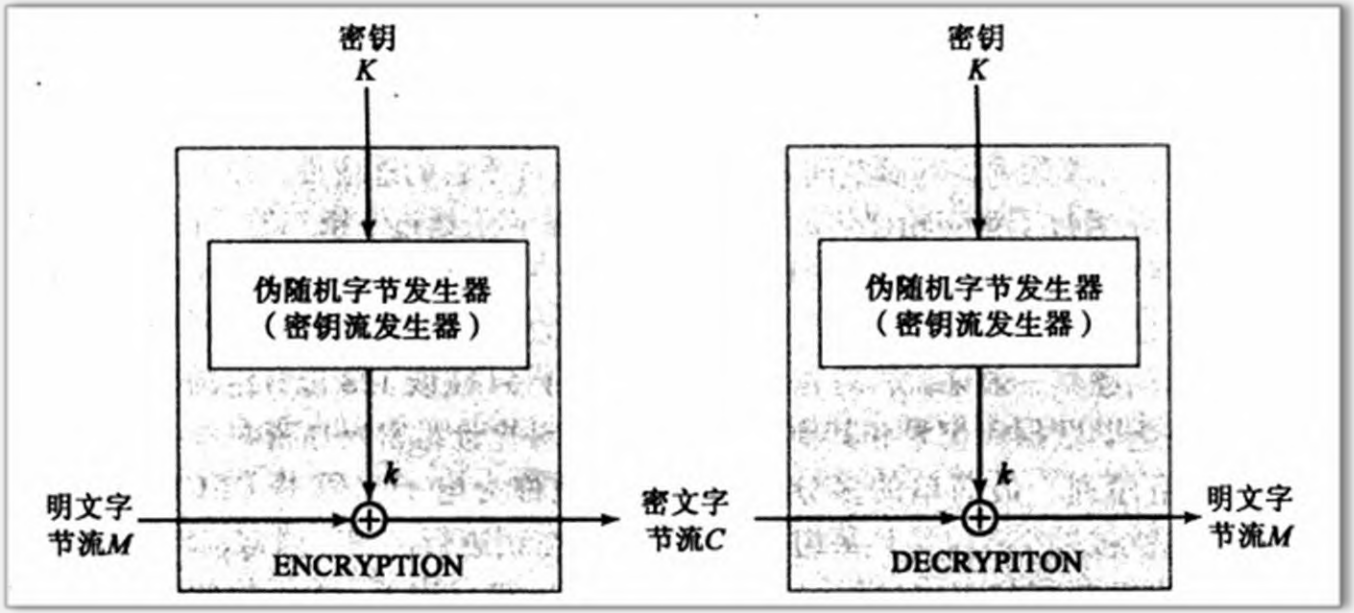
- 因 3DES 自身有缺陷不能长期作为理想加密算法标准，NIST 在 1997 年公开征集新的 高级加密标准（AES），要求其安全强度不低于 3DES 且提高计算效率等，规定 AES 必须是分组长度为 128 位的对称分组密码，并支持 128 位、192 位和 256 位的密钥。
- 1999 年有 15 个候选算法通过第一轮评估，仅 5 个通过第二轮，2001 年 11 月 NIST 完成评估并发布最终标准（FIPS PUB 197），选择 Rijndael 作为建议的 AES 算法，AES 已广泛应用于商业产品。

	DES	3DES	AES
明文分组长度（位）	64	64	128
密文分组长度（位）	64	64	128
密钥长度（位）	56	112 或 168	128、192 或 256

2.1 流密码

- 分组密码一次处理输入的一个元素分组，每个输入分组产生一个输出分组；流密码持续处理输入元素，每次产生一个元素的输出。
- 流密码相对分组密码的主要优点是速度更快且编写代码更少，分组密码优点是可重复使用密钥。
  - 对于数据流加密 / 解密的应用，如数据通信信道、浏览器 / Web 连接通信，流密码是好方案。
  - 对于处理成块数据的应用，如文件传输、电子邮件和数据库，分组密码更适用
  - 实际中两种密码都可用于各种应用。

2.1 流密码



这是一个典型的流密码结构图。在该结构中，密钥输入到一个伪随机位发生器中，该伪随机位发生器产生一串随机的 8 位数。一个伪随机流就是在不知道输入密钥的情况下不可预知的流，其明显具有随机特性。发生器的输出称为密钥流（keystream），每次组合成一个字节，并与明文流进行逐位异或（XOR）运算。

2.2 消息认证和散列函数

- 当消息、文件、文档或其他数据集合是真实的且来自于合法信源，则被称为是可信的。
- 消息认证是一种允许通信者验证所接收或存储的数据是否可信的措施。
- 认证包括两个重要方面：验证消息的内容有没有被篡改和验证信源是否可信。
- 还可以验证消息的时效性（即消息没有被人为地延迟和重放）以及两个实体之间传输的消息流的相对顺序。

2.2 利用对称加密实现认证

- 仅仅简单地使用对称加密也可以进行消息认证：
  - 假设只有发送方和接收方共享一个密钥（这是合理的），那么只有真正的发送方才能够成功地为对方加密消息，提供接收者能识别的有效消息。
  - 如果消息里带有检错码和序号，则接收方就可以确认消息是否被修改过以及顺序是否正确。
  - 如果消息还包括时间戳，那么接收方就可以确认消息的传输有没有超出网络传输的正常延迟。
- 但事实上，单独的对称加密并不是数据认证的有效工具。

2.2 无须加密的消息认证

- 消息本身没有提供一个安全的认证方式。
- 通过加密消息加上认证标签可以将认证和机密性组合在一个算法中。
- 三种采用无机密性的消息认证更为恰当的情形:
  - 将同一消息广播到很多目的地。比如通知各用户网络暂时不可使用或者控制中心发出的报警信号等。
  - 在信息交换中，通信一方处理负荷较大，不能承担解密收到的所有消息的时间开销。
  - 对明文形式的计算机程序进行认证是一种很吸引人的服务。

2.2 消息认证码（MAC）

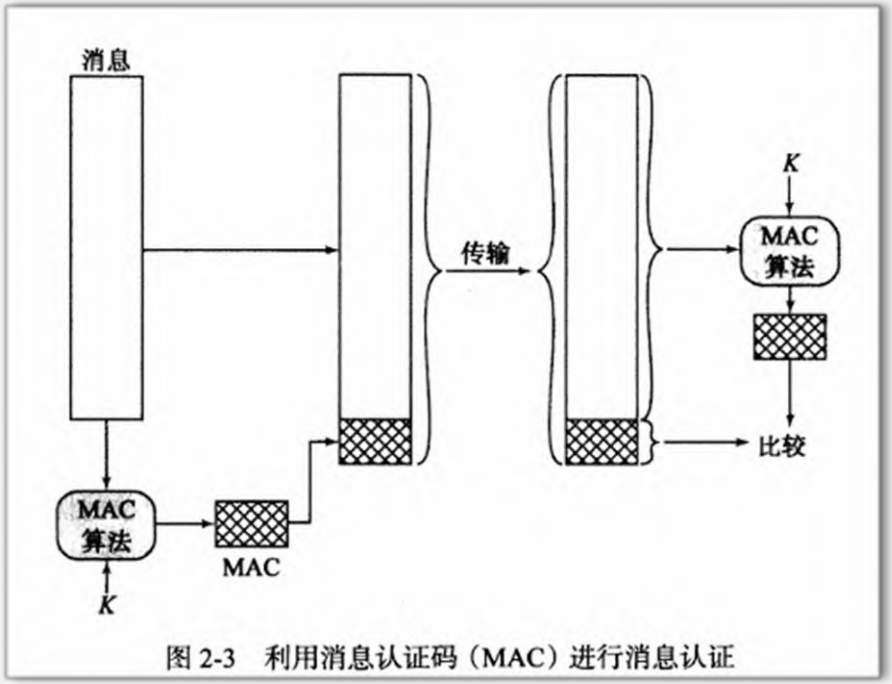


图 2-3　利用消息认证码（MAC）进行消息认证

消息认证码（MAC）技术：一种利用秘密密钥生成固定长度短数据块的认证技术。

-假设只有收发双方知道秘密密钥

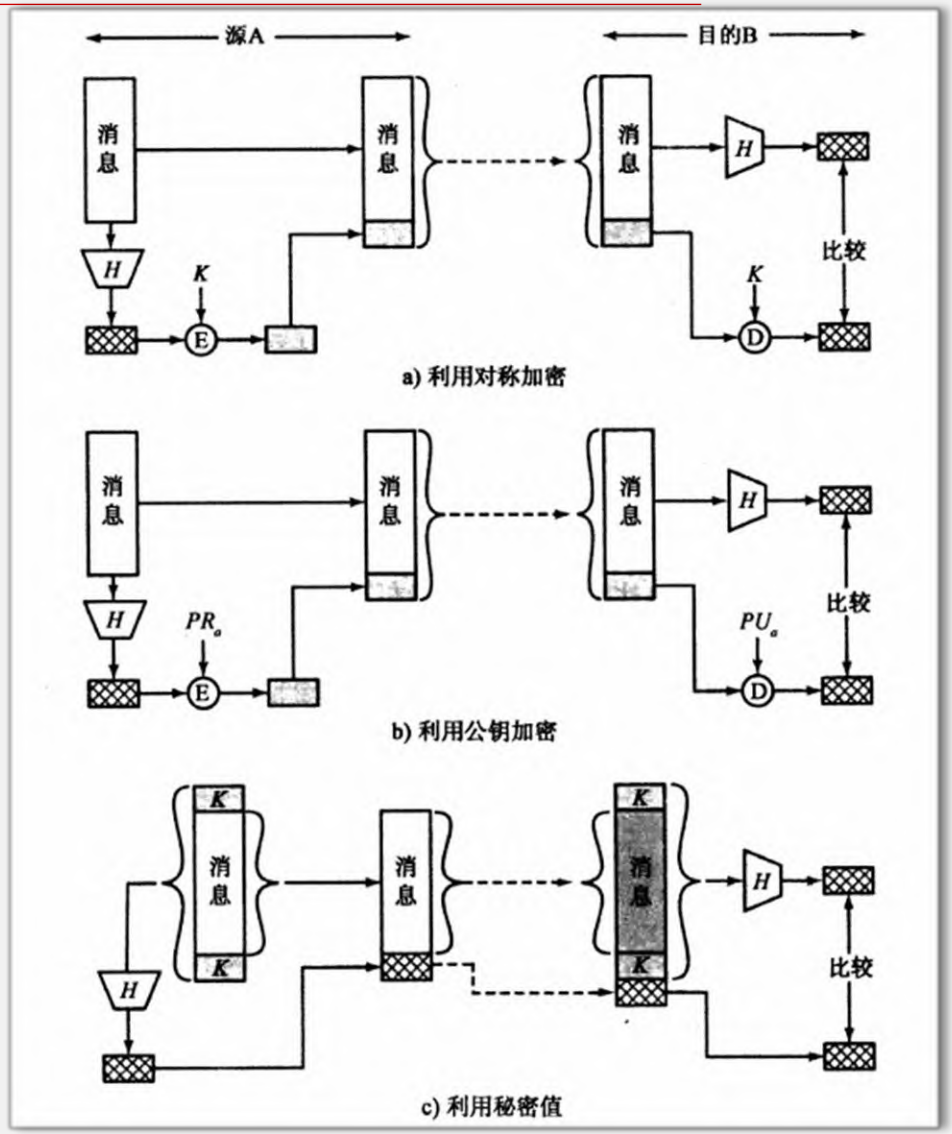
1. 通信双方A和B使用共同秘密密钥K
2. A向B发送消息时计算消息认证码MAC=F(K,M)
3. 消息和认证码一起发送给接收方
4. 接收方用相同密钥计算新的消息认证码并与收到的认证码比较
5. 若相等则表明消息可信

2.2 单向散列函数

- 单向散列函数是消息认证码的一种变形。
- 与消息认证码一样，散列函数接受可变长度的消息M作为输入，产生固定长度的消息摘要H(M)作为输出。
- 一般来说，消息的长度被填充到某个固定长度（如1024位）的整数倍，填充的消息包括原始消息y位为单位的长度值。这个长度字段作为一个安全措施增加了攻击者利用散列值改变消息的难度。

2.2 三种消息认证的方式

- 利用对称加密。假定只有发送方和接收方共享加密密钥，那么可信性就可以得到保证。
- 利用公钥加密。利用公钥密码加密有两个优点:其提供了数字签名和消息认证;不要求密钥分发到通信各方。
- 利用散列函数但不使用加密的消息认证技术。这项技术被称为密钥散列MAC。



2.2 对散列函数的要求

- 散列函数的目的就是要产生文件、消息或其他数据块的“指纹”。一个散列函数H要能够用于消息认证，它必须具有下列性质:
  1. H可应用于任意大小的数据块。
  2. H产生固定长度的输出。
  3. 对任意给定的x，计算H(x)比较容易，用硬件和软件均可实现。
  4. 对任意给定的散列码h，找到满足H(X)=h的x在计算上是不可行的。具有这种性质的散列函数被称为是单向的或抗原象的。
  5. 对任意给定的分组x，找到满足y≠x 且H(y)=H(x)的y 在计算上是不可行的，具有这种性质的散列函数被称为是第二抗原象，有时也被称为是弱抗碰撞的。
  6. 找到任何满足H(x)=H(y)的偶对(x,y)在计算上是不可行的。具有这种性质的散列函数被称为是抗碰撞的。有的也被称为是强抗碰撞的。

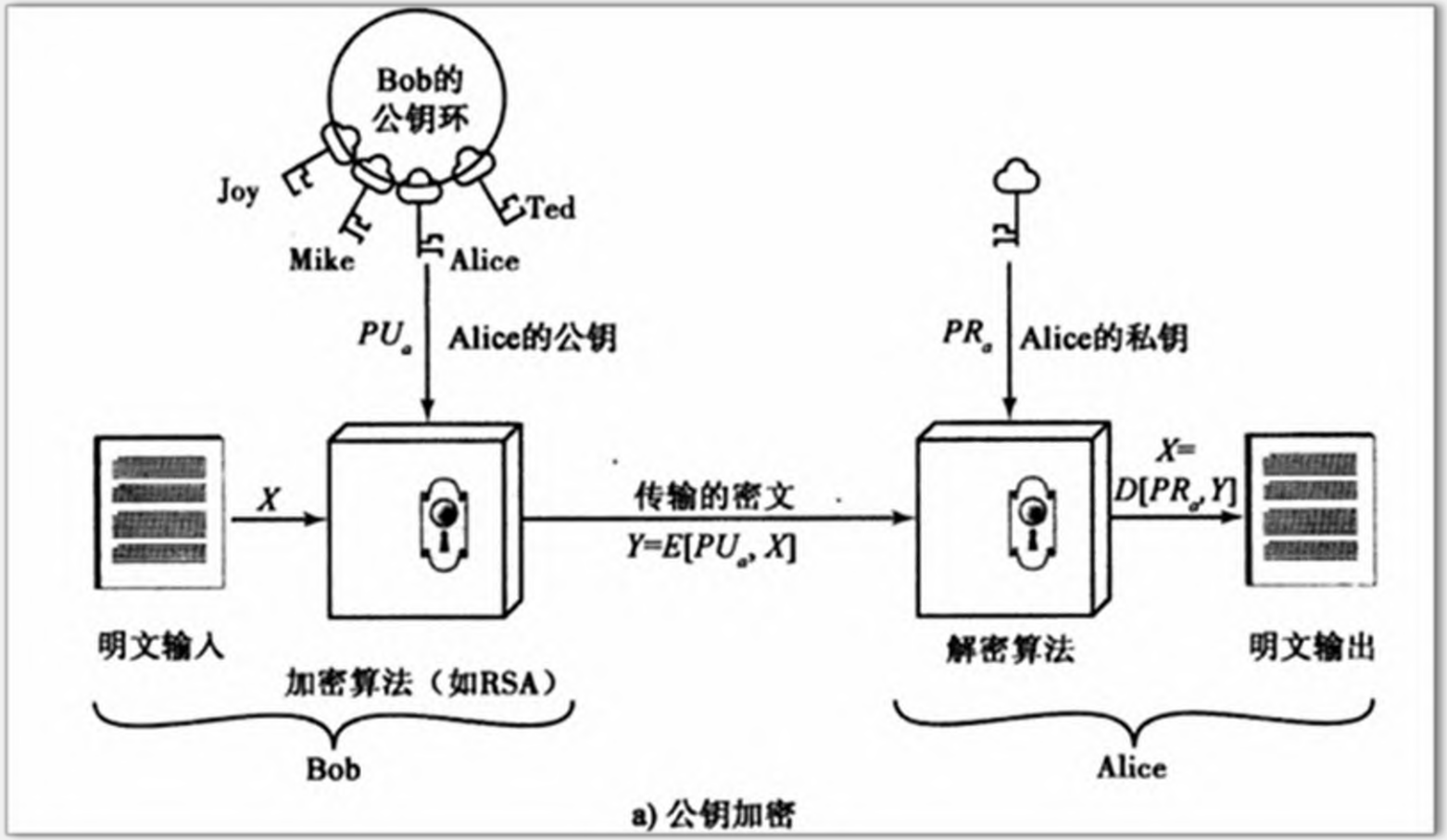
2.2 安全散列函数

- 和对称加密一样，对安全散列函数的攻击也有两类:密码分析和蛮力攻击。
- 在最近几年里，安全散列算法（Secure Hash Algorithm，SHA)已成为使用最广泛的散列算法。
  - SHA 由NIST设计并于1993年作为联邦信息处理标准（FIPS 180)发布。
  - 当SHA 的缺点被发现后，修订版于1995年发布(FIPS 180-1)，通常称之为SHA-1。SHA-1产生160位的散列值。
  - 2002年，NIST提出了该标准的修订版（FIPS 180-2)，它定义了三种新的SHA版本，散列值的长度分别为256位、384位和512位，分别称为SHA-256,SHA-384和SHA-512。
  - 这些新的版本统称为SHA-2，它们与SHA-1具有相同的基本结构，并使用了相同类型的模运算和逻辑二元运算。特别是512位的SHA-2版本，似乎具有牢不可破的安全性。
  - 然而，鉴于SHA-2与SHA-1两者结构的相似性，NIST决定标准化一种与SHA-2和SHA-1截然不同的新的散列函数。这种被称为SHA-3的新散列函数发布于2012年，而且现在已经作为SHA-2的替代品而被广泛运用。

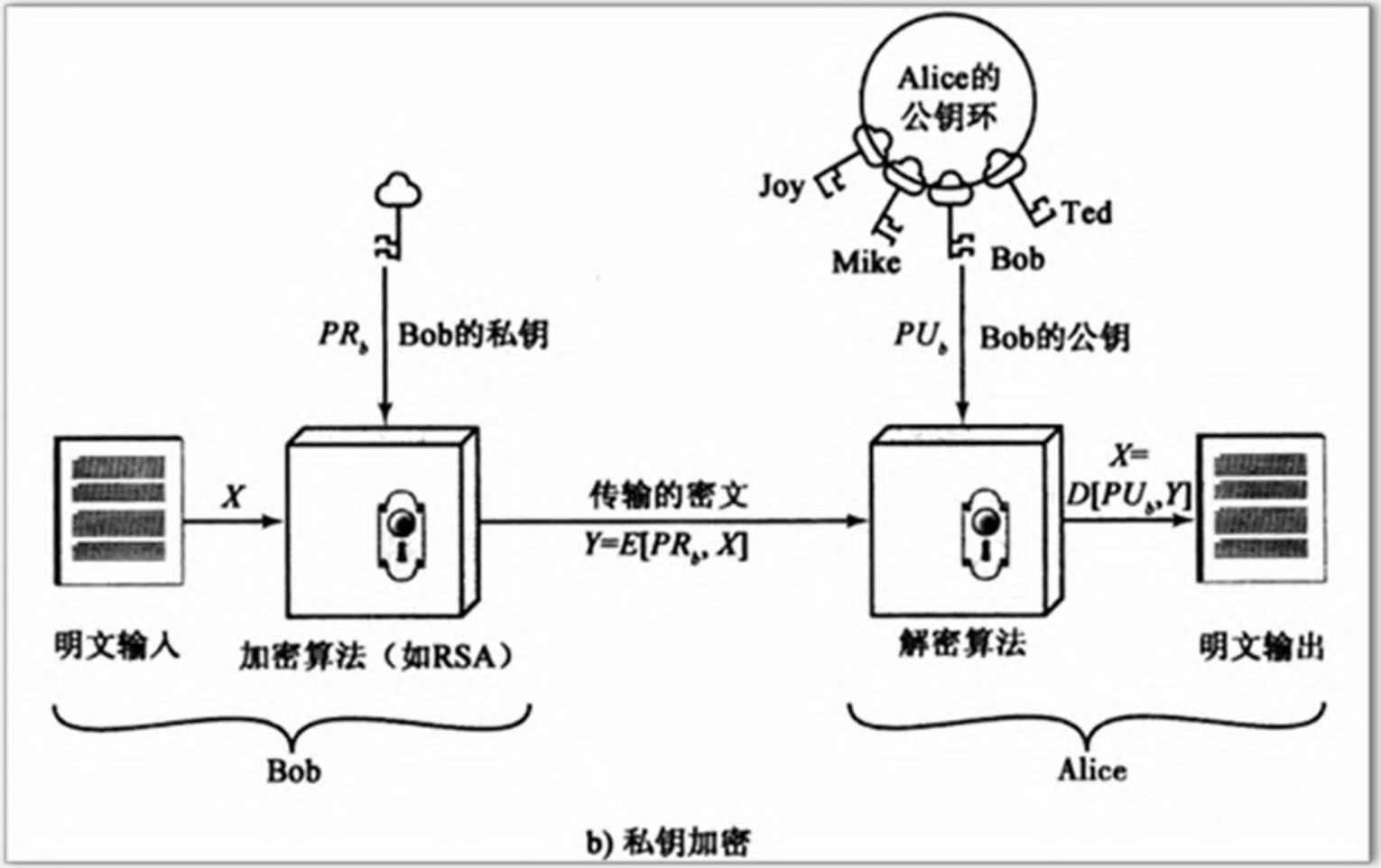
2.3 公钥加密

- 1976年，Diffie和Hellman【DIFF76】首次提出了公钥加密的思想，这是有文字记载的几千年来秘密领域第一次真正革命性的进步。
- 公钥算法基于数学函数，而不像对称加密算法那样是基于位模式的简单操作，更重要的是，公钥密码是非对称的(asymmetric）它使用两个单独的密钥。
- 而对称加密只是用一个密钥。使用两个密钥对于机密性、密钥分发和认证产生了意义深远的影响。
- 公钥对其他使用者来说是公共的。然而私钥只有它的拥有者知道。
- 一般的公钥加密算法依赖于一个加密密钥和一个与之不同但又相关的解密密钥。

2.3 公钥加密


Bob的
公钥环
Joy
Mike
Alice
Ted
PU_a
Alice的公钥
PR_a
Alice的私钥
X
传输的密文
$Y=E[PU_a, X]$
$X=D[PR_a, Y]$
明文输入
加密算法（如RSA）
Bob
解密算法
明文输出
Alice
a) 公钥加密

2.3 公钥加密


Alice的
公钥环
Joy
Mike
Bob
Ted
PR_b Bob的私钥
PU_b Bob的公钥
X
传输的密文
$Y=E[PR_b, X]$
$X=D[PU_b, Y]$
明文输入
加密算法（如RSA）
解密算法
明文输出
Bob
Alice
b) 私钥加密

2.3 对公钥密码的要求

① B产生一对密钥（公钥PU，私钥PR）计算容易。

② 已知公钥和要加密的消息M，A产生相应的密文应计算容易：C = E(PU, M)。

③ B用私钥对接收的密文解密恢复明文计算容易：M = D(PR, C) = D[PR, E(PU, M)]。

④ 已知公钥PU时，攻击者确定私钥PR在计算上不可行。

⑤ 已知公钥PU和密文C，攻击者恢复明文M在计算上不可行。

⑥ 还可增加条件，加密和解密函数顺序可交换：M = D[PU, E(PR, M)] = D[PR, E(PU, M)]。

2.3 非对称加密算法

- MIT的Ron Rivest、Adi Shamir和Len Adleman在1977年提出了第一个公钥体制RSA，并于1978年首次发表该算法。RSA被认为是使用最广泛并被实现的公钥加密方法。RSA体制是一种分组密码，其明文和密文均是0～n-1之间的整数。
- Diffie和 Hellman在一篇具有独创意义的论文中首次提出了公钥算法，给出了公钥密码学的定义，该算法被称为Diffie-Hellman密钥交换或密钥协议。许多商业产品都使用了这种密钥交换技术。该算法的目的是使两个用户能安全地交换密钥，以便在后续的通信中用该密钥对消息加密。该算法本身只限于进行密钥交换。

2.3 非对称加密算法

- 美国国家标准与技术研究所(NIST)发布的联邦信息处理标准FIPS PUB186，被称为数字签名标准（Digital Signature Standard，DSS)。DSS使用SHA-1算法给出了一种新的数字签名方法，即数字签名算法（DSA）。DSS使用的是只提供数字签名功能的算法。与RSA不同，它不能用于加密和密钥分发。
- 大多数使用公钥密码学和数字签名的产品和标准都使用RSA算法。为了保证RSA 的使用安全性，最近这些年来密钥的位数一直在增加，这对使用RSA的应用是很重的负担，对于进行大量安全交易的电子商务更是如此。最近，一种具有强大竞争力的椭圆曲线密码学（Elliptic Curve Cryptography，ECC)对RSA提出了挑战。在标准化的过程中，如关于公钥密码学的IEEE(电气和电子工程师学会)P1363标准中，人们也已考虑了ECC.

2.3 公钥密码体制的应用

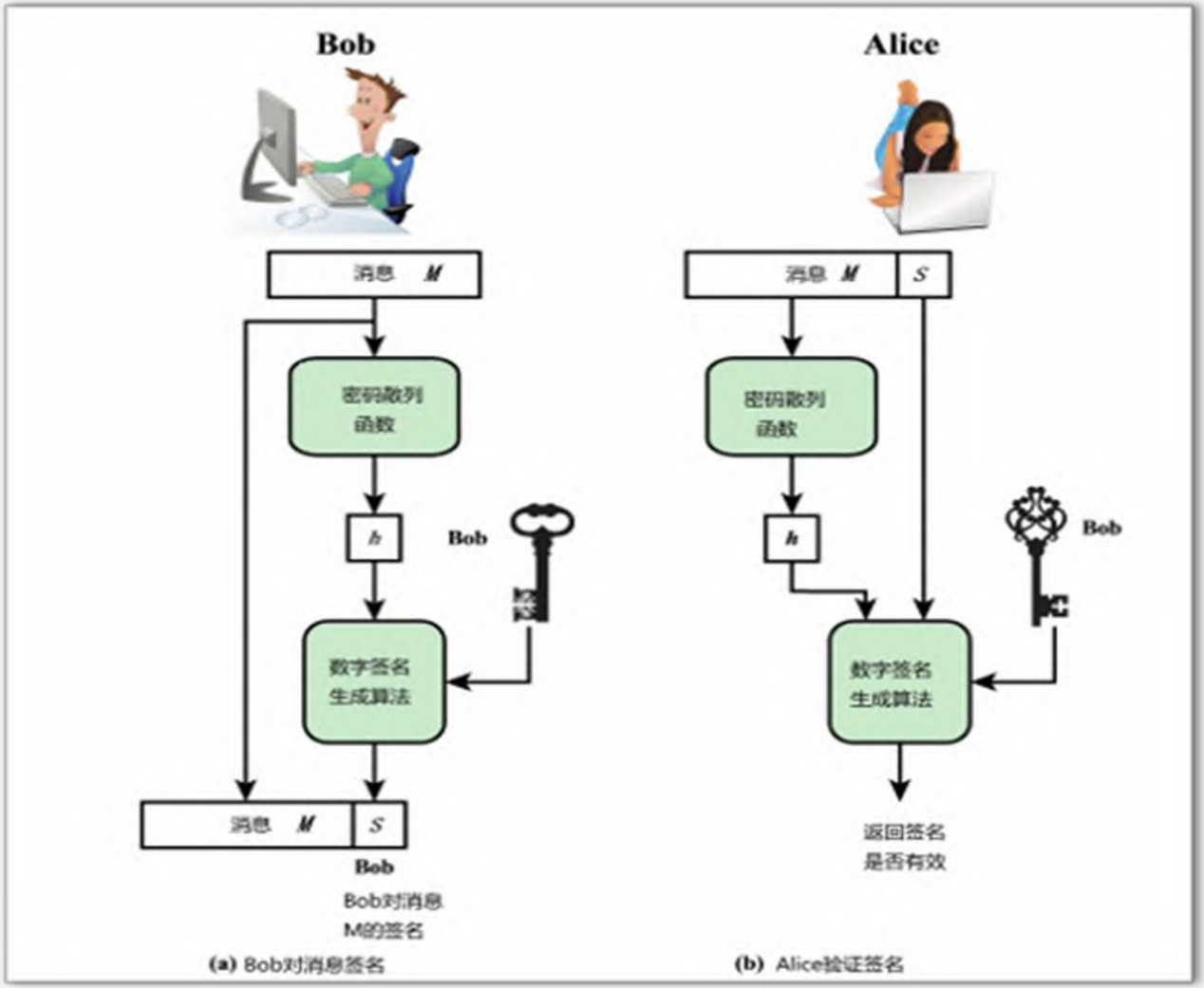
- 公钥密码体制的特点是使用具有两个密钥的密码算法，其中一个密钥是私有的，另一个是公有的。
- 根据不同的应用，发送方可使用其私钥或者接收方的公钥或同时使用两者来实现密码功能。
- 一般来讲，可以把公钥密码体制的应用划分为三类：数字签名、对称密钥分发和秘密密钥加密。

算　法	数字签名	对称密钥的分发	秘密密钥加密
RSA	是	是	是
Diffie-Hellman	否	是	否
DSS	是	否	否
椭圆曲线	是	是	是

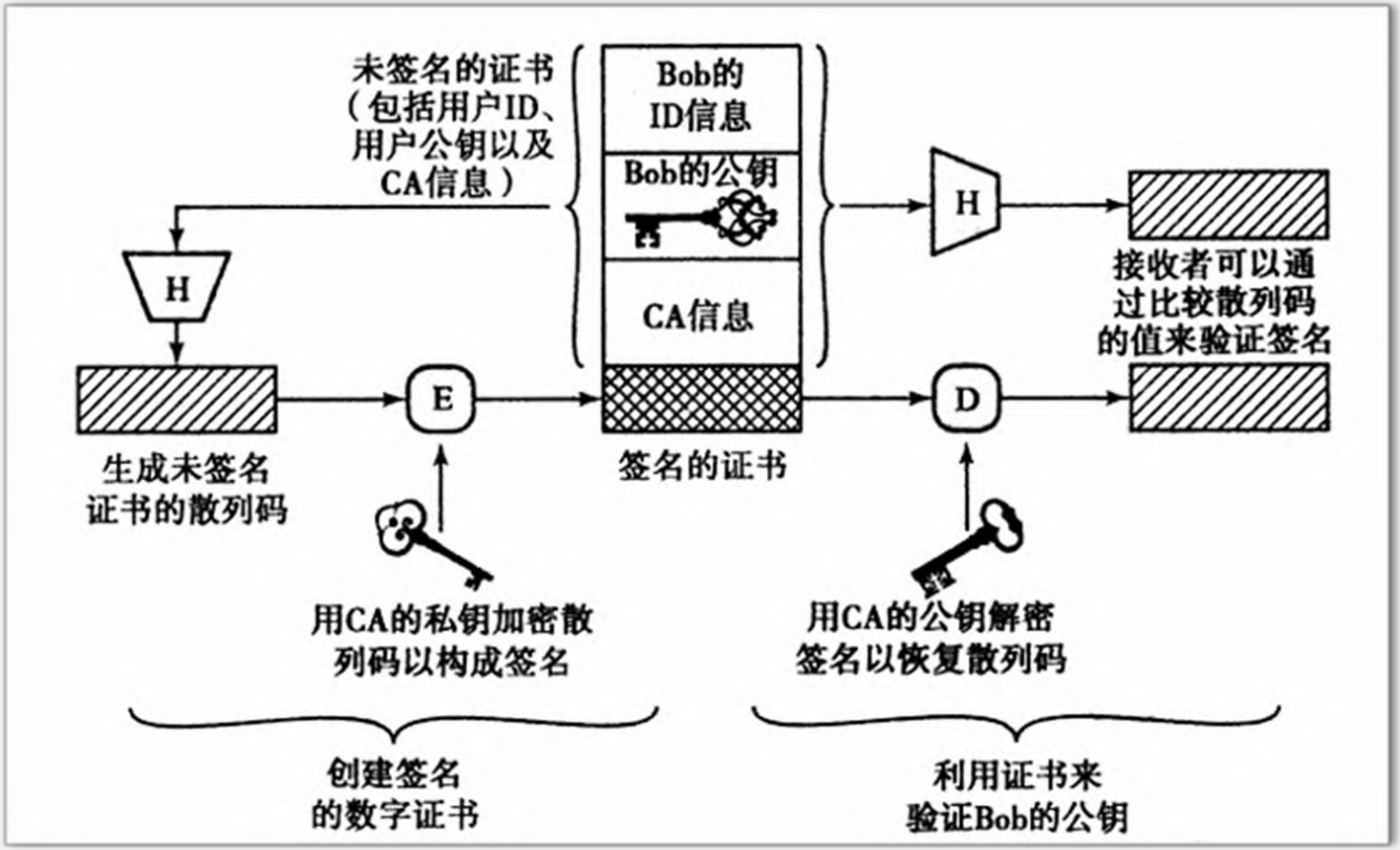
2.4 数字签名

- NIST FIPS PUB 186-4 定义了数字签名:
  - “数据的加密转换的结果（如果得到适当实施）能够提供一个机制来保证原始认证、数据完整性以及签名的不可依赖性。”
- 数字签名是依赖于数据的位组合格式，由代理根据文件、消息或其他形式的数据块生成
- FIPS 186-4 指定了以下三种数字签名算法:
  1. 数字签名算法(DSA)
  2. RSA 数字签名算法
  3. 椭圆曲线数字签名算法(ECDSA)

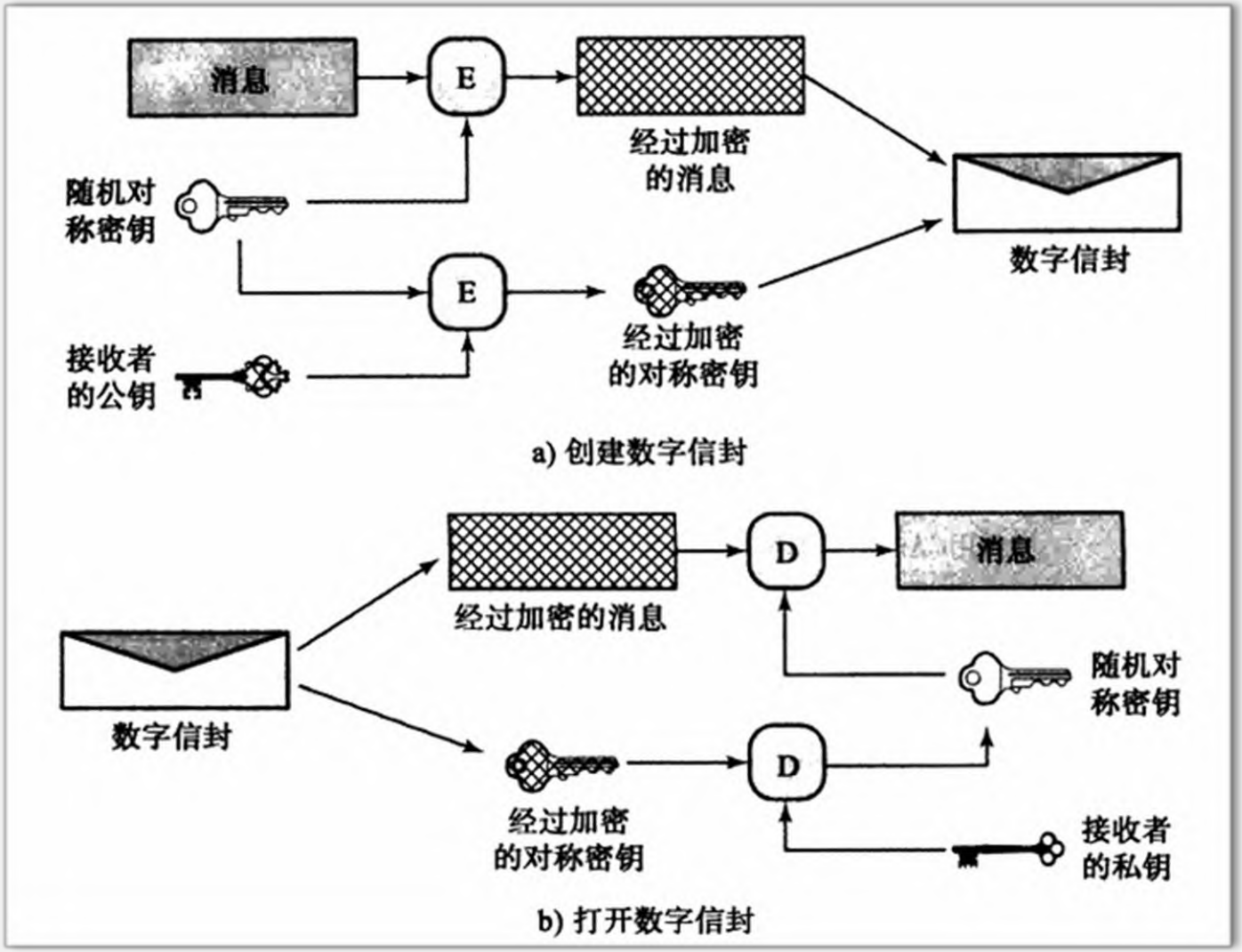
2.4 数字签名


Bob
消息 M
密码散列函数
h
Bob
数字签名生成算法
消息 M
S
Bob
Bob对消息M的签名
(a) Bob对消息签名
Alice
消息 M
S
密码散列函数
h
Bob
数字签名生成算法
返回签名是否有效
(b) Alice验证签名

2.4 公钥证书


未签名的证书
（包括用户ID、
用户公钥以及
CA信息）
Bob的
ID信息
Bob的公钥
CA信息
H
H
接收者可以通
过比较散列码
的值来验证签名
E
D
生成未签名
证书的散列码
签名的证书
用CA的私钥加密散
列码以构成签名
用CA的公钥解密
签名以恢复散列码
创建签名
的数字证书
利用证书来
验证Bob的公钥

2.4 数字信封


消息
E
经过加密
的消息
随机对
称密钥
数字信封
E
经过加密
的对称密钥
接收者
的公钥
a) 创建数字信封
D
消息
经过加密的消息
随机对
称密钥
数字信封
D
经过加密
的对称密钥
接收者
的私钥
b) 打开数字信封

2.5 随机数的使用

- 大量基于密码学的网络安全算法使用了随机数，例如:
  - RSA公钥加密算法和其他公钥算法中密钥的产生。
  - 对称流密码中流密钥的生成。
  - 用作临时会话密钥或产生数字信封的对称密钥的生成。
  - 在许多密钥分发环境如Kerberos域中，随机数作为握手信号以防止重放攻击。
  - 会话密钥的生成，无论其是由密钥分发中心完成还是由一个本体(principal)完成。
- 这些应用对随机数序列提出了两种截然不同且不一定兼容的要求:随机性和不可预测性。

2.5 随机与伪随机

- 密码应用大多使用算法来生成随机数。这些算法是确定的，所以产生的序列并非是统计随机的。不过要是算法好的话，产生的序列可以经受住随机性检测。这样的数一般被称为伪随机数。
- 在大多数情况下，伪随机数在实际应用中会表现得想真随机数一样。尽管“像真随机数一样”这种说法是非常主观的，然而伪随机数已被普遍接受。
- 一个真随机数发生器是利用不确定的源来生成真随机数。大部分是通过测量不可预测的自然过程（如电离辐射效应的脉冲检测器、气体排放管和漏电电容器）来实现的。

2.6 实际应用：存储数据的加密

- 计算机系统的一条基本安全要求是保护存储数据。对存储数据进行保护的安全机制包括访问控制、入侵检测和入侵保护策略。除了技术方面的因素外，一些人为因素也可以使这些方法变得脆弱。
- 2005年4月，San Jose医疗机构宣称有人通过物理途径偷走了它们的一台计算机并可能获得185 000条未加密的病人记录。
- 便携电脑丢失的例子不计其数:在机场丢失，在停放着的汽车中被盗，在用户离开办公桌时被拿走。如果存储在这些便携电脑硬盘上的数据未加密，则所有的数据都将被窃贼获得。

2.6 实际应用：存储数据的加密

- 尽管现在商业上已经对通过网络、Internet或无线设备传输的信息提供了包括加密在内的各种保护手段，可一旦数据存储在本地（称为静态数据(data at rest)），就很少有域认证和操作系统访问控制之外的保护了。
- 因此，对静态数据加密并结合有效的加密密钥管理体制的方案变得颇具吸引力，实际上应该强制执行。
- 有许多方法可以用来进行加密。
  - 利用一个商用的加密包
  - 后端器件
  - 基于库的磁带加密
  - 便携电脑和PC后台数据加密