

第2章 密码学基础

韩 琦

基本信息

- 课程目标

- 全面了解信息安全**知识领域的体系结构**，了解信息安全发展历史，了解信息安全技术产生与发展，了解信息安全问题对国家及社会的影响；
- 理解密码的**基本理论及应用方法**，掌握身份认证及访问控制的基本方法，具有设计信息安全方案的初步能力，了解**网络威胁**的产生机理与**网络防御**的基本技术，具有**解决信息安全问题**的初步能力；
- 了解信息安全系统、工程的**技术标准及操作规范**，明确有关信息安全的法律法规，建立初步信息安全工程素质和全面的信息安全**法律意识**。

主要内容

2.1 密码学基础知识

2.2 古典密码

2.3 对称密钥密码

2.4 公开密钥密码

2.5 消息认证

2.6 密码学新进展

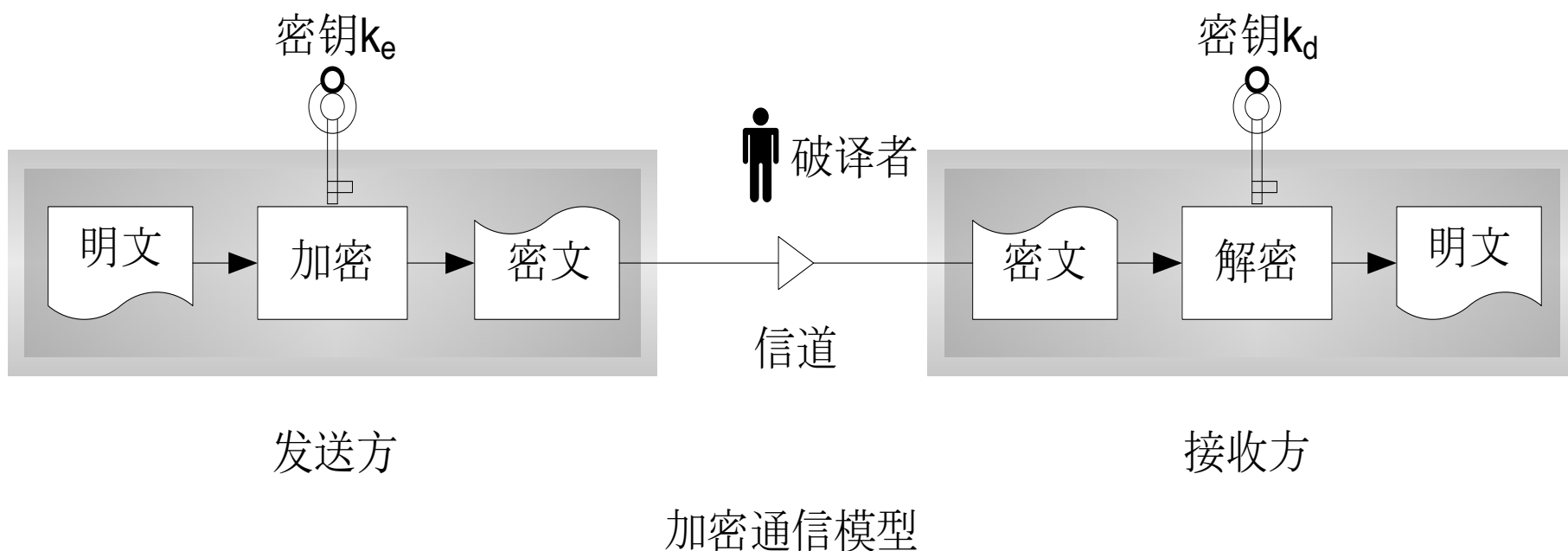
2.1 密码学基础知识

2.1.1 引言

- 解决数据的机密性、完整性、不可否认性以及身份识别等问题均需要以密码为基础
 - 密码技术是保障信息安全的核心基础。
- 密码学(Cryptography)包括密码编码学和密码分析学两部分。
 - 将密码变化的客观规律应用于编制密码用来保守通信秘密的，称为密码编码学；
 - 研究密码变化客观规律中的固有缺陷，并应用于破译密码以获取通信情报的，称为密码分析学。

2.1.2 密码体制

- 消息在密码学中被称为**明文** (Plain Text)
- 伪装消息以隐藏它的内容的过程称为**加密** (Encrypt)
- 被加密的消息称为**密文** (Cipher Text)
- 把密文转变为明文的过程称为**解密** (Decrypt)



2.1.2 密码体制

— 完整密码体制要包括如下五个要素

- M 是可能明文的有限集称为明文空间;
- C 是可能密文的有限集称为密文空间;
- K 是一切可能密钥构成的有限集称为密钥空间;
- E 为加密算法, 对于任一密钥, 都能够有效地计算;
- D 为解密算法, 对于任一密钥, 都能够有效地计算。

— 密码体系必须满足如下特性:

- 加密算法($E_k: M \rightarrow C$)和解密算法($D_k: C \rightarrow M$)满足:
 $D_k(E_k(x))=x$, 这里 $x \in M$;
- 破译者不能在有效的时间内破解出密钥 k 或明文 x 。

2.1.3 密码的分类

— 依据密码体制的特点以及出现的时间分类:

- 古典密码
- 现代密码
 - 对称密钥密码
 - 公开密钥密码

— 依据处理数据的类型

- 分组密码(block cipher)
- 序列密码(stream cipher)

2.2 古典密码

古典密码

— 什么是古典密码？

- 古代人用的密码？
- 1940年代之前，安全性依赖于算法保密的密码方法

— 古典密码的分类线索

- 置换密码
- 代换密码
 - 单表代换
 - 多表代换

2.2.1 置换密码

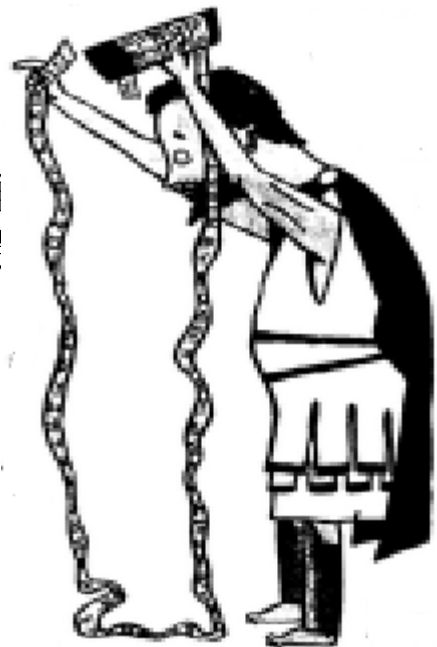
最早的密码应用实例：置换密码

背景：斯巴达军队逐渐占据了优势地位，准备对雅典发动最后一击。原来站在斯巴达一边的波斯帝国突然改变态度，停止了对斯巴达的援助，意图是雅典和斯巴达在持续的战争中两败俱伤，以便从中渔利。

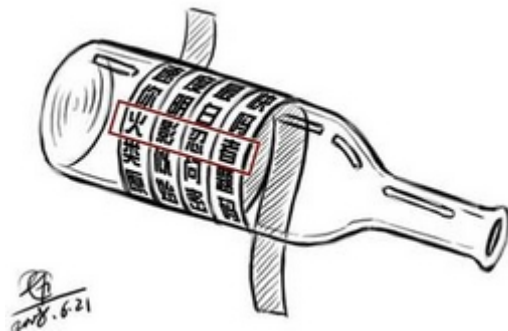


2.2.1 置换密码

截获信使：斯巴达军队俘获了一名从波斯帝国回雅典送信的雅使，仔细搜查后只有一条布满杂乱无章的希腊字母的普通腰带，他别无他获。



破译密码：斯巴达军队统帅莱桑德把腰带上的字母用各种方法重新排列组合，怎么也解不出来。最后，当他无意中把腰带呈螺旋形缠绕在手中的剑鞘上时，奇迹出现了。原来腰带上杂乱无章的字母，竟组成了一段文字。



2.2.2 凯撒（Caesar）密码

- 凯撒密码的数学表达：
 - 将字母表的字母右移 k 个位置，并对字母表长度作模运算：
 - 加密函数： $E_k(m) = (m + k) \bmod q$
 - 解密函数： $D_k(c) = (c - k) \bmod q$
- 举例：
 - $M=C=\{\text{有序字母表}\}$, $q = 26$, $k = 3$ 。
 - 明文信息 $M = \text{meet me after the toga party}$
 - 密文信息 $C = \text{phhw ph diwho wkh wrjd sduwb}$

2.2.3 乘数密码

- 乘数密码

- 将明文字母串逐位乘以密钥 k 并进行模运算。

- 数学表达式: $E_k(m) = k \times m \bmod q$, $\gcd(k, q) = 1$ 。

- 算法描述:

- $M=C=Z/(26)$, 明文空间和密文空间同为英文字母表空间, 包含26个元素; $q=26$;
 - $K=\{k \in \text{整数集} \mid 0 < k < 26, \gcd(k, 26)=1\}$, 密钥为大于0小于26, 与26互素的正整数;
 - $E_k(m) = k \times m \bmod q$ 。
 - $D_k^{-1}(c) = k^{-1} \times c \bmod q$, 其中 k^{-1} 为 k 在模 q 下的乘法逆元。

2.2.4 仿射密码

- 仿射密码
 - 可以看作是移位密码和乘数密码的结合。
- 算法描述：
 - $M=C=Z/(26)$; $q=26$;
 - $K=\{k_1, k_2 \in Z \mid 0 < k_1, k_2 < 26, \gcd(k_1, 26)=1\}$;
 - $E_k(m)=(k_1 m + k_2) \bmod q$;
 - $D_k(c)=k_1^{-1}(c - k_2) \bmod q$, 其中 k_1^{-1} 为 k_1 在模 q 下的乘法逆元。

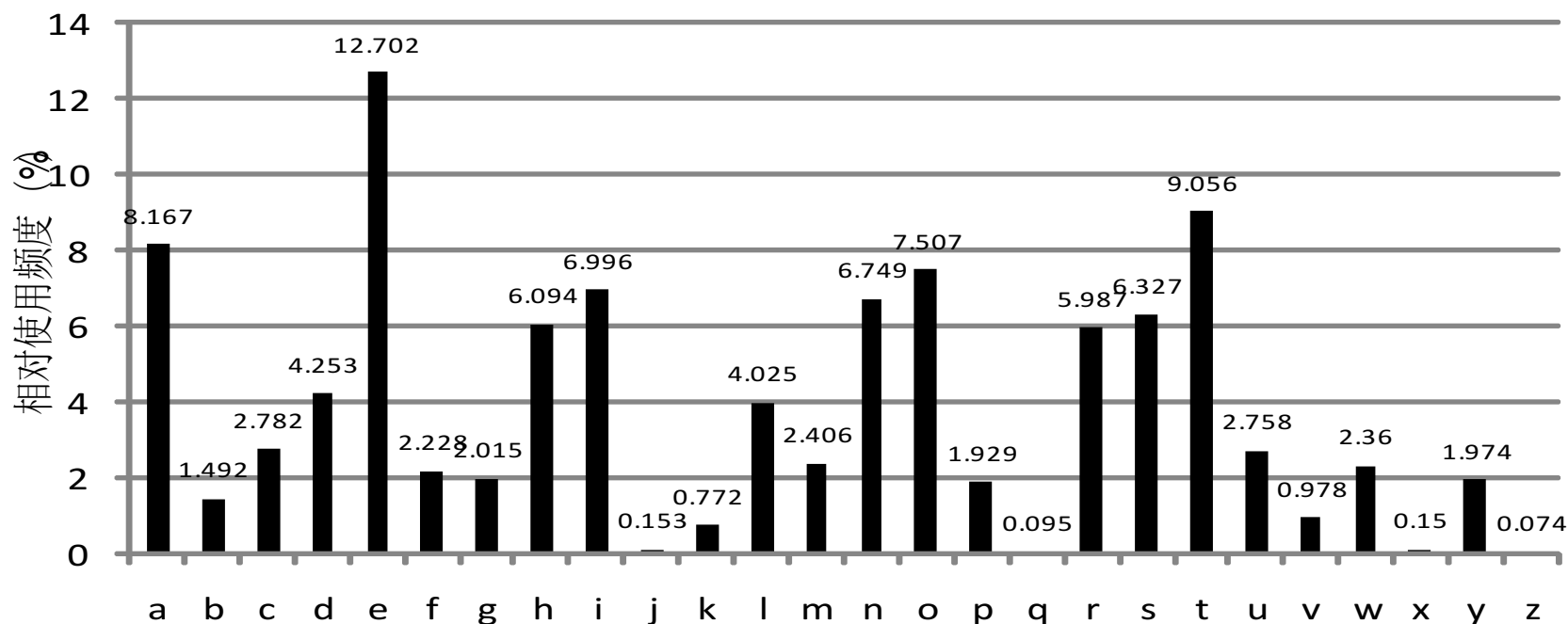
2.2.4 仿射密码

- 设 $k = (5, 3)$ ，注意到 $5^{-1} \bmod 26 = 21$,
- 加密函数:
 - $E_k(x) = 5x + 3 \pmod{26}$,
- 解密函数:
 - $D_k(y) = 21(y - 3) \bmod 26 = 21y - 11 \pmod{26}$ 。
- 加密明文 “yes” 的加密与解密过程如下:

$$\begin{array}{c}
 E_k \left\{ \begin{matrix} y \\ e \\ s \end{matrix} \right\} = 5 \times \left\{ \begin{matrix} 24 \\ 4 \\ 18 \end{matrix} \right\} + \left\{ \begin{matrix} 3 \\ 3 \\ 3 \end{matrix} \right\} = \left\{ \begin{matrix} 19 \\ 23 \\ 15 \end{matrix} \right\} = \left\{ \begin{matrix} t \\ x \\ p \end{matrix} \right\} \\
 \leftarrow \text{Mod 26} \rightarrow \\
 \leftarrow \text{加密过程} \rightarrow
 \end{array}
 \quad
 \begin{array}{c}
 21 \times \left\{ \begin{matrix} 19 \\ 23 \\ 15 \end{matrix} \right\} - \left\{ \begin{matrix} 11 \\ 11 \\ 11 \end{matrix} \right\} = \left\{ \begin{matrix} 24 \\ 4 \\ 18 \end{matrix} \right\} = \left\{ \begin{matrix} y \\ e \\ s \end{matrix} \right\} \\
 \leftarrow \text{Mod 26} \rightarrow \\
 \leftarrow \text{解密过程} \rightarrow
 \end{array}$$

2.2.5 针对古典密码的攻击

- 简单代替密码的加密是从明文字母到密文字母的一一映射
- 攻击者统计密文中字母的使用频度，比较正常英文字母的使用频度，进行匹配分析。
- 如果密文信息足够长，很容易对单表代替密码进行破译。



2.2.6 多表代换密码

- 多表代换密码是以一系列代换表依次对明文消息的字母进行代换的加密方法。
- 多表代换密码使用从明文字母到密文字母的多个映射来隐藏单字母出现的频率分布。
- 经典的多表代换密码：
 - **Vigenère**、Beaufort、Running Key、Vernam和**轮转机**等密码。
- 非周期多表代换密码（表的数量 $\rightarrow\infty$ ）
 - 对每个明文字母都采用不同的代替表(或密钥)进行加密，称作**一次一密密码**。

2.2.6 多表代替密码

- 维吉尼亚（Vigenère）密码

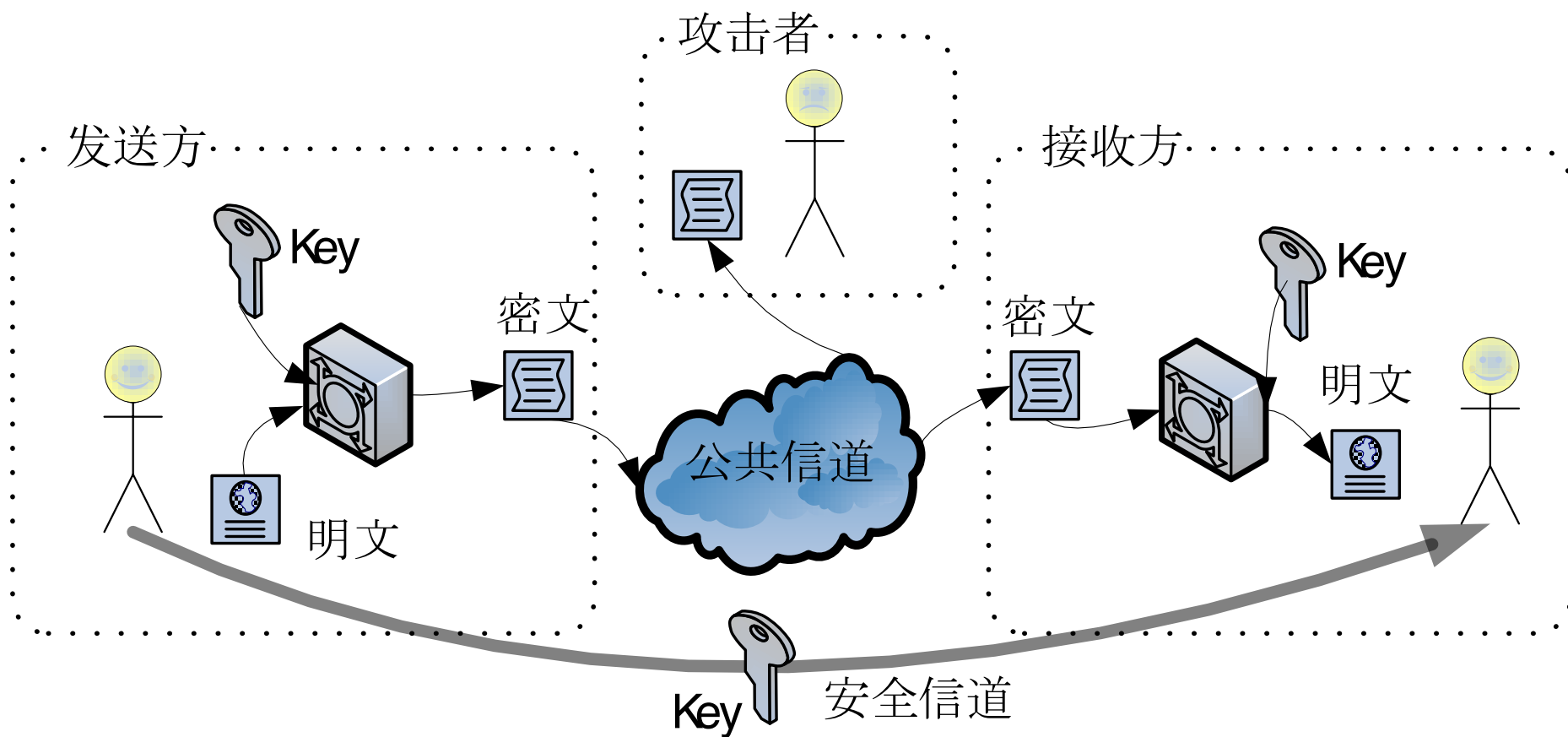
- 特点：

- 是以移位代替为基础的周期多表代替密码。
 - 加密时每一个密钥被用来加密一个明文字母，当所有密钥使用完后，密钥又重新循环使用。

- 算法：

- $E_k(m) = C_1 C_2 \dots C_n$ ，其中 $C_i = (m_i + k_i) \bmod 26$;
 - 密钥K可以通过周期性反复使用以至无穷。

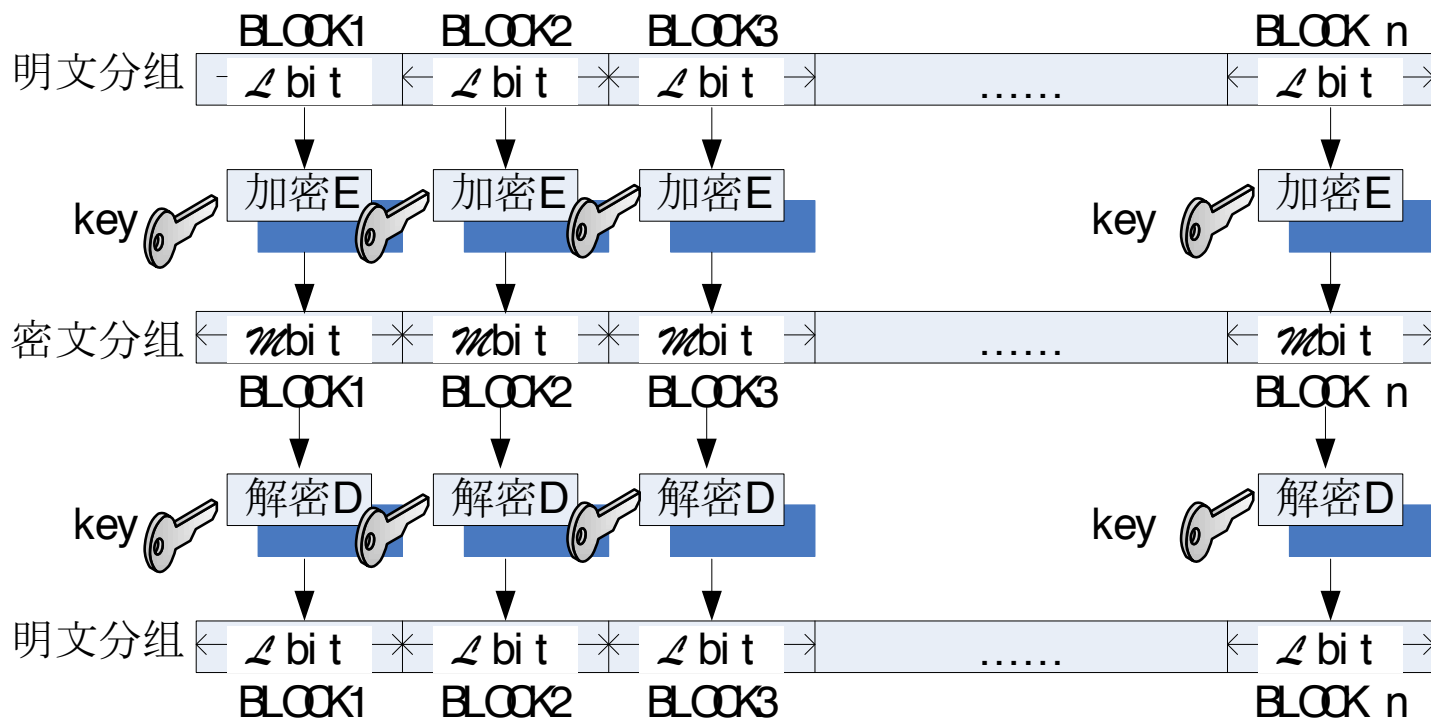
2.3 对称密钥密码



对称密钥密码的模型

2.3.1 对称密钥密码加密模式

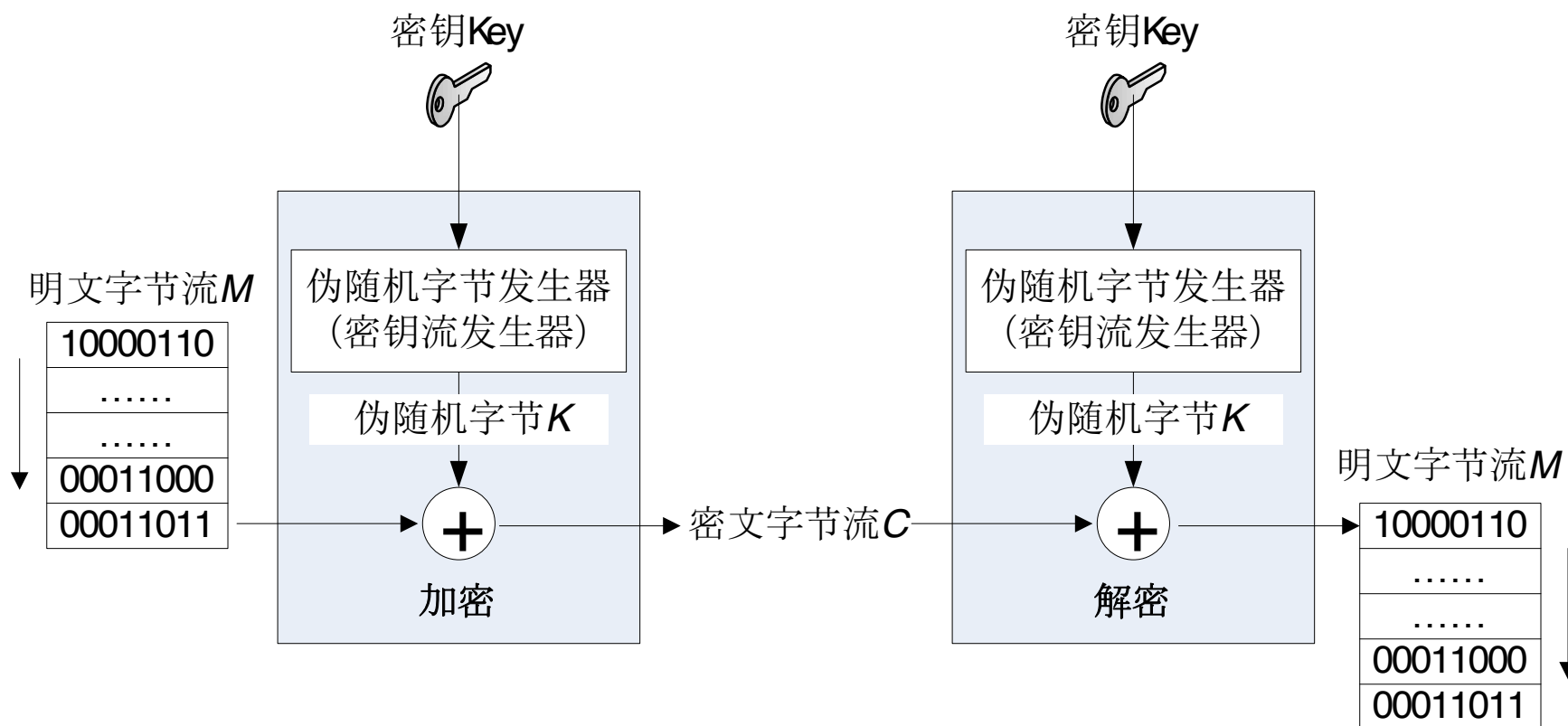
- 分组密码：
 - 明文消息分成若干固定长度的组，进行加密；解密亦然。



分组密码工作原理示意图

2.3.1 对称密钥密码加密模式

- 序列密码（流密码）：
 - 通过伪随机数发生器产生性能优良的伪随机序列(密钥流)，用该序列加密明文消息流，得到密文序列；解密亦然。



序列密码工作原理示意图

- [在线DES加密解密](#)
- [DES加密算法详解](#)
- [在线加密解密算法](#)

2.3.2 数据加密标准DES

– 1973年美国国家标准局NBS公开征集国家密码标准方案；

- ① 算法必须提供高度的安全性；
- ② 算法必须有详细的说明，并易于理解；
- ③ 算法的安全性取决于密钥，不依赖于算法；
- ④ 算法适用于所有用户；
- ⑤ 算法适用于不同应用场合；
- ⑥ 算法必须高效、经济；
- ⑦ 算法必须能被证实有效；
- ⑧ 算法必须是可出口的。

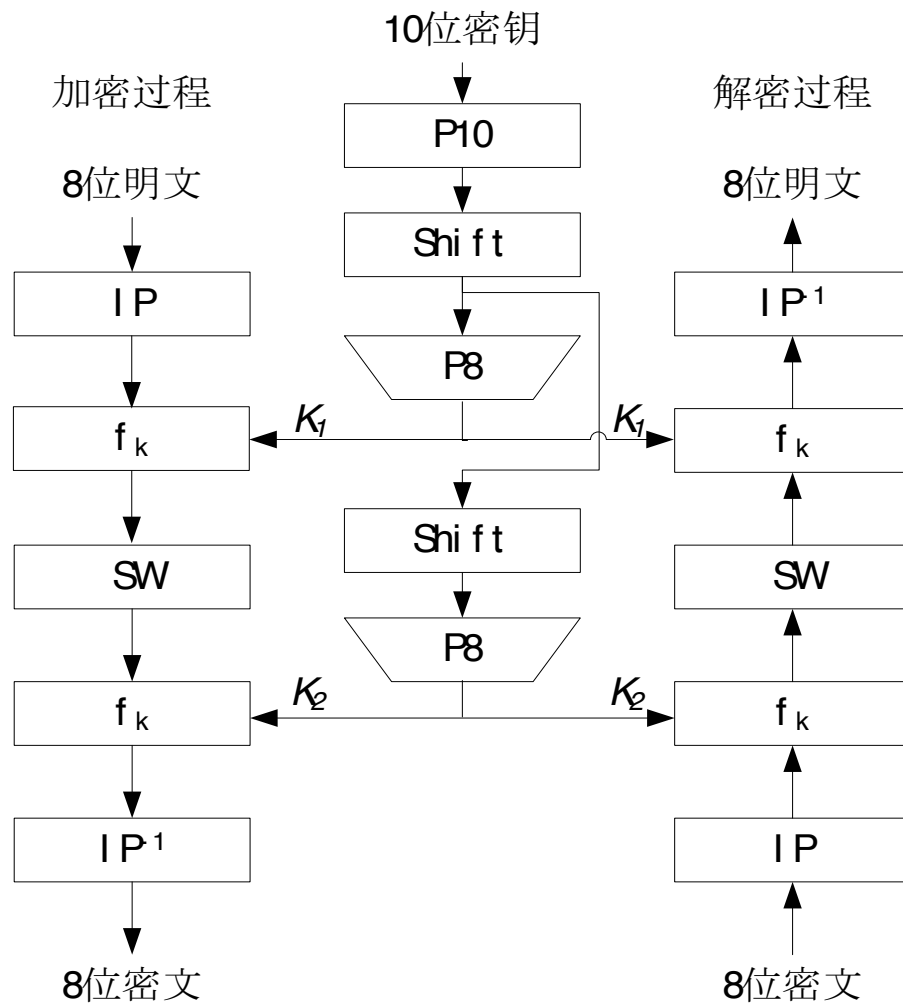
– 1974年NBS开始第二次征集时，IBM公司提交了算法LUCIFER。

- 1977年LUCIFER被美国国家标准局NBS作为“数据加密标准 FIPS PUB 46”发布，简称为DES

S-DES

- S-DES加密算法

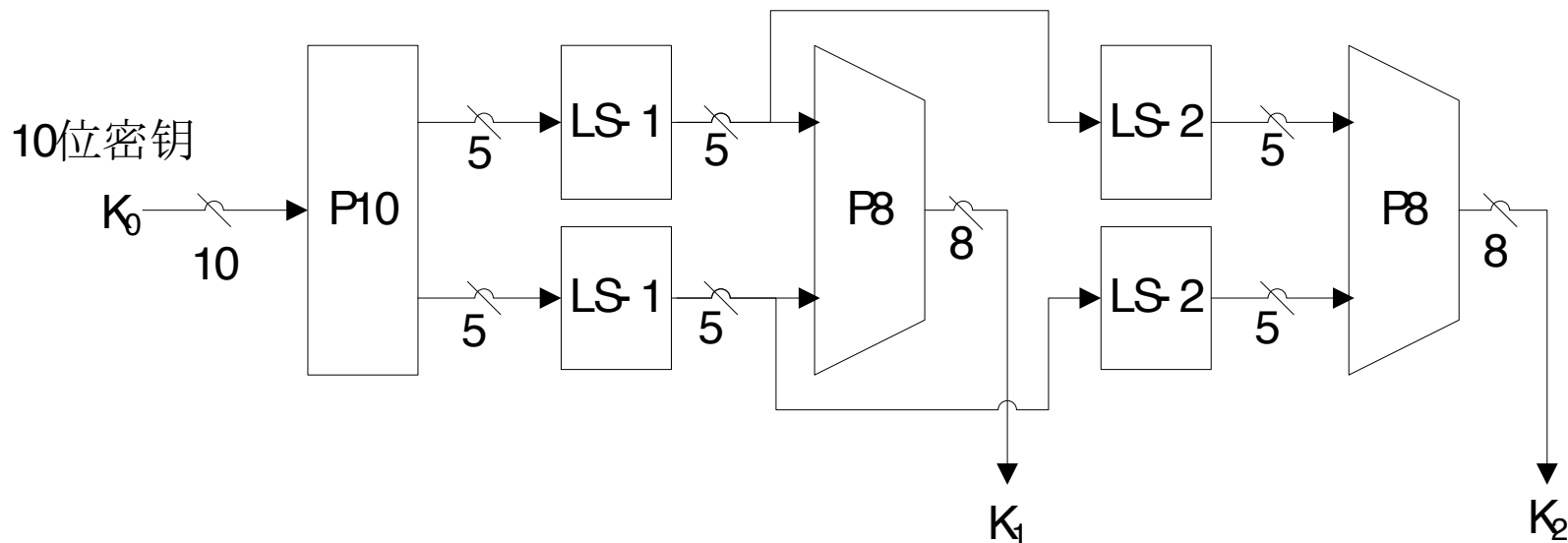
- S-DES是由美国圣达卡拉大学的Edward Schaeffer教授提出的，主要用于教学，其设计思想和性质与DES一致，有关函数变换相对简化，具体参数要小得多。
- 输入为一个8位的二进制明文组和一个10位的二进制密钥，输出为8位二进制密文组；
- 解密与加密基本一致。
- 加密： $IP^{-1}(f_{k_2}(SW(f_{k_1}(IP(\text{明文}))))))$
- 解密： $IP^{-1}(f_{k_1}(SW(f_{k_2}(IP(\text{密文}))))))$



S-DES的体制

S-DES的密钥产生

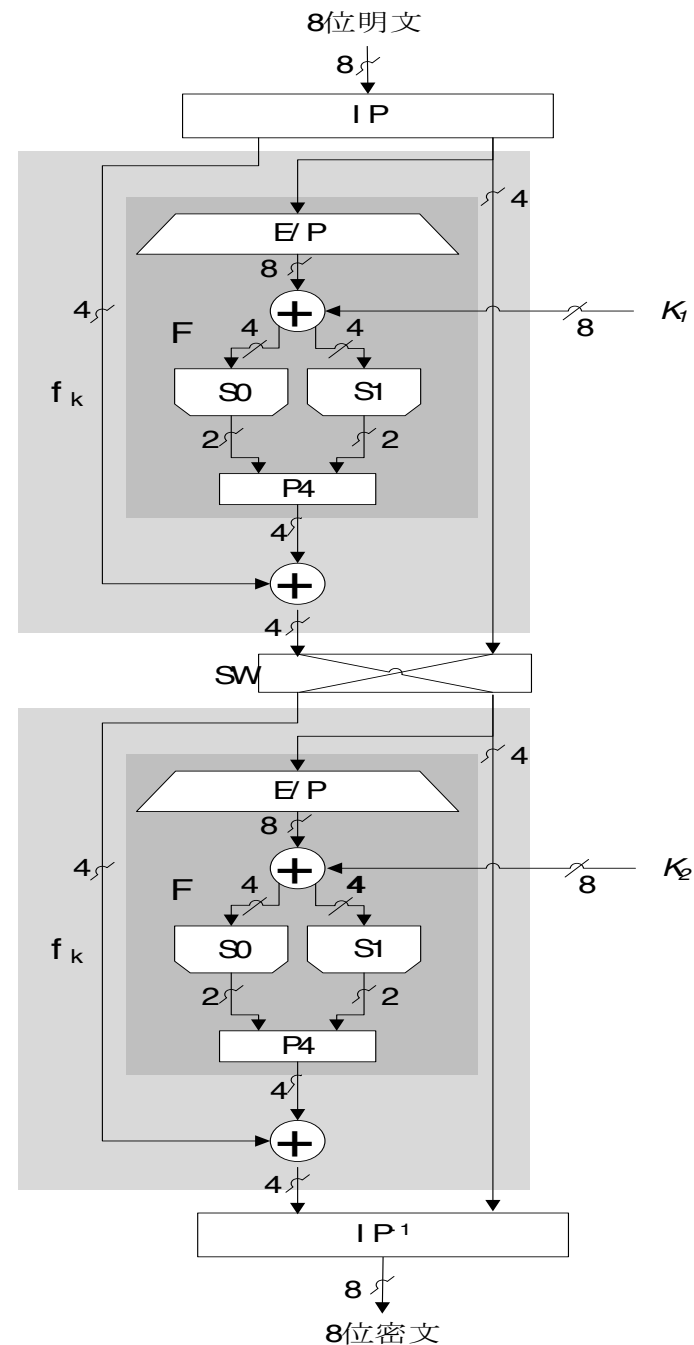
- $P_{10}=(3,5,2,7,4,10,1,9,8,6)$
- 循环左移函数LS
- $P_8=(6,3,7,4,8,5,10,9)$



S-DES的密钥产生

S-DES的加密变换过程

- $IP=(2,6,3,1,4,8,5,7)$;
- $IP^{-1}=(4,1,3,5,7,2,8,6)$
- $E/P=(4,1,2,3,2,3,4,1)$
- “ \oplus ”:按位异或运算;
- $P4=(2,4,3,1)$
- S盒函数
 - $S0$ 和 $S1$ 为两个盒子函数, 将输入作为索引查表, 得到相应的系数作为输出。
- SW :将左4位和右4位交换。



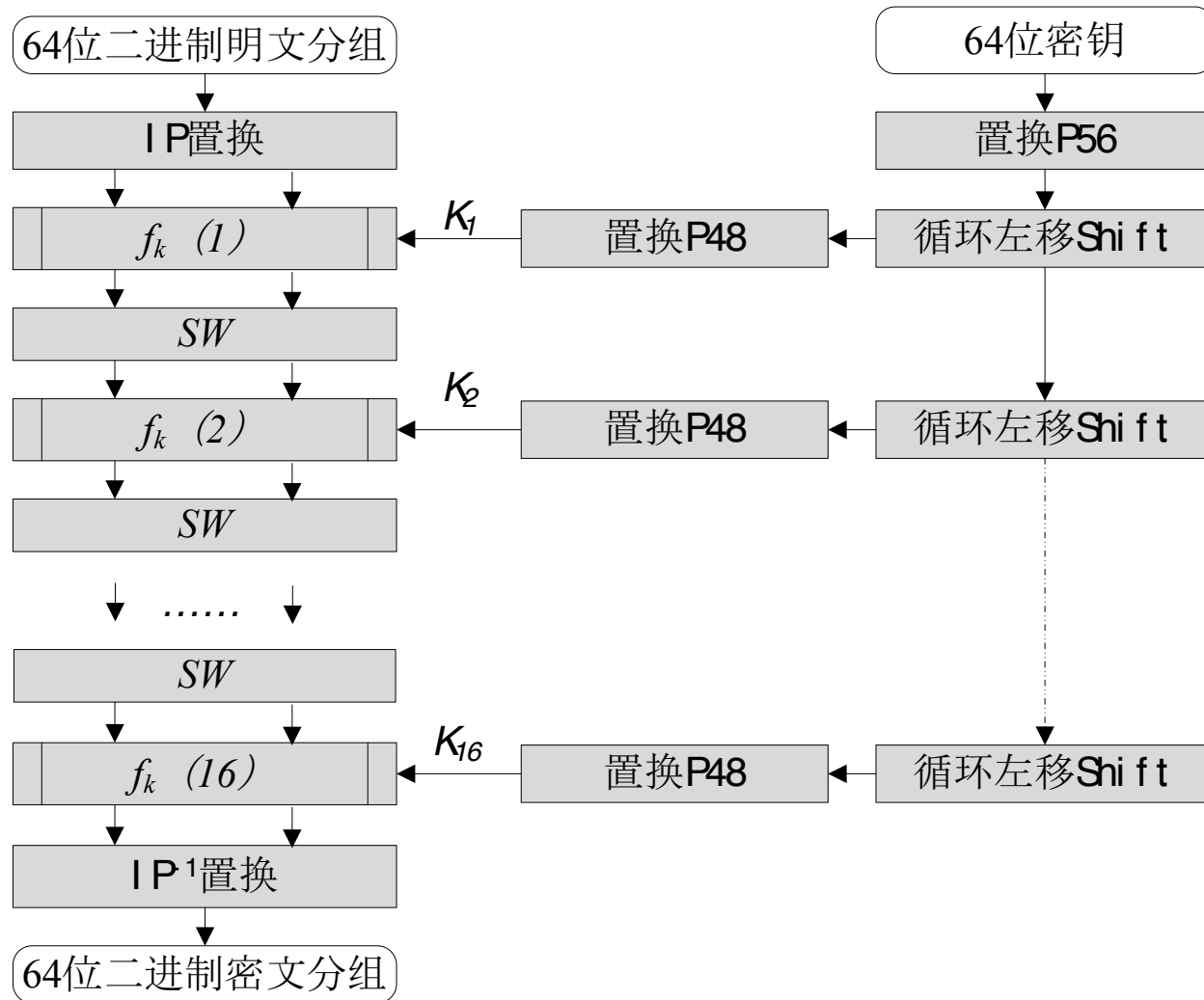
S-DES的加密过程

S盒函数

$$S_0 = \begin{matrix} & 0 & 1 & 2 & 3 \\ \begin{matrix} 0 \\ 1 \\ 2 \\ 3 \end{matrix} & \begin{bmatrix} 1 & 0 & 3 & 2 \\ 3 & 2 & 1 & 0 \\ 0 & 2 & 1 & 3 \\ 3 & 1 & 3 & 2 \end{bmatrix} \end{matrix} \quad S_1 = \begin{matrix} & 0 & 1 & 2 & 3 \\ \begin{matrix} 0 \\ 1 \\ 2 \\ 3 \end{matrix} & \begin{bmatrix} 0 & 1 & 2 & 3 \\ 2 & 0 & 1 & 3 \\ 3 & 0 & 1 & 0 \\ 2 & 1 & 0 & 3 \end{bmatrix} \end{matrix}$$

- S盒函数按下述规则运算：
 - 输入的第1位和第4位二进制数合并为一个两位二进制数，作为S盒的行号索引*i*；
 - 将第2位和第3位同样合并为一个两位二进制数，作为S盒的列号索引*j*，
 - 确定S盒矩阵中的一个系数 (*i*, *j*) 。
 - 此系数以两位二进制数形式作为S盒的输出。
 - 例如：
 - $L' = (l_0, l_1, l_2, l_3) = (0, 1, 0, 0)$, $(i, j) = (0, 2)$
 - 在S₀中确定系数3，则S₀的输出为11B。

DES算法



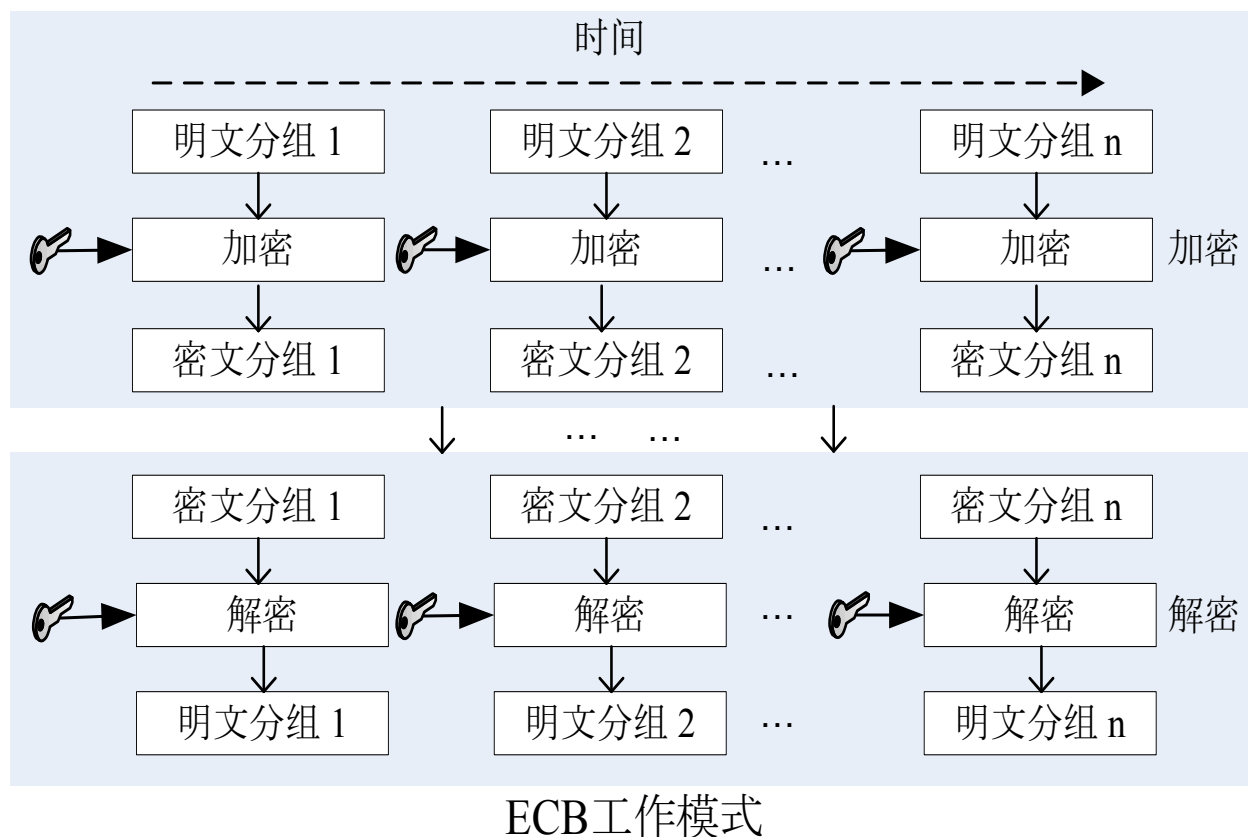
DES算法框图

DES的安全问题

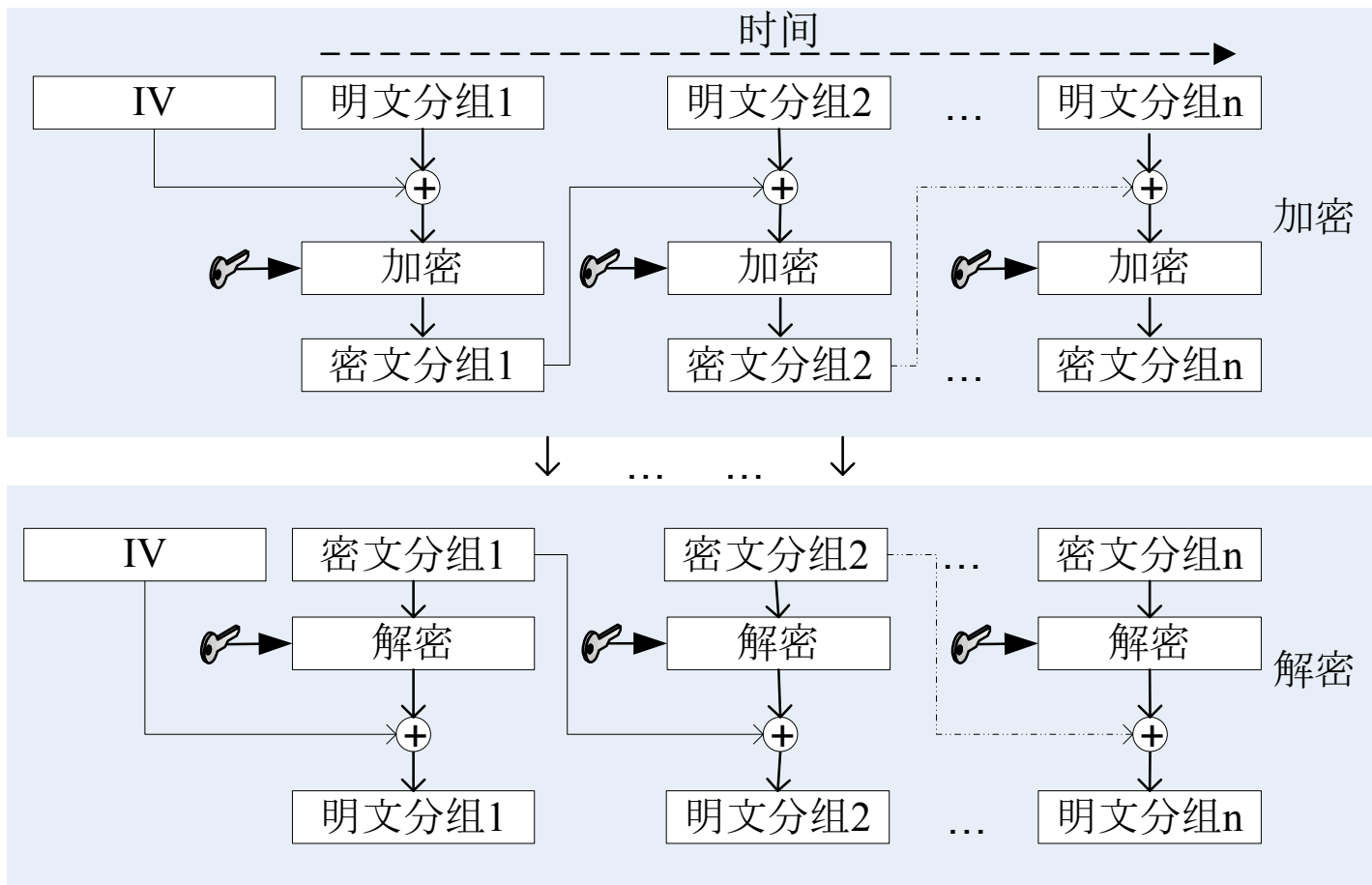
- 1977年，耗资两千万美元建成一个专门计算机用于DES的破译，需要12个小时的破解才能得到结果。
- 1994年世界密码大会，M.Matsui提出线性分析方法，利用243个已知明文，成功破译DES，
- 1997年首届“向DES挑战”的竞技赛。罗克·维瑟用了96天时间破解了用DES加密的一段信息。
- 2000年1月19日，电子边疆基金会组织25万美元的DES解密机以22.5小时成功破解DES加密算法。
- DES的最近一次评估是在1994年，同时决定1998年12月以后，DES将不再作为联邦加密标准。

2.3.3 分组密码的工作模式

- 电子编码本模式（ECB）

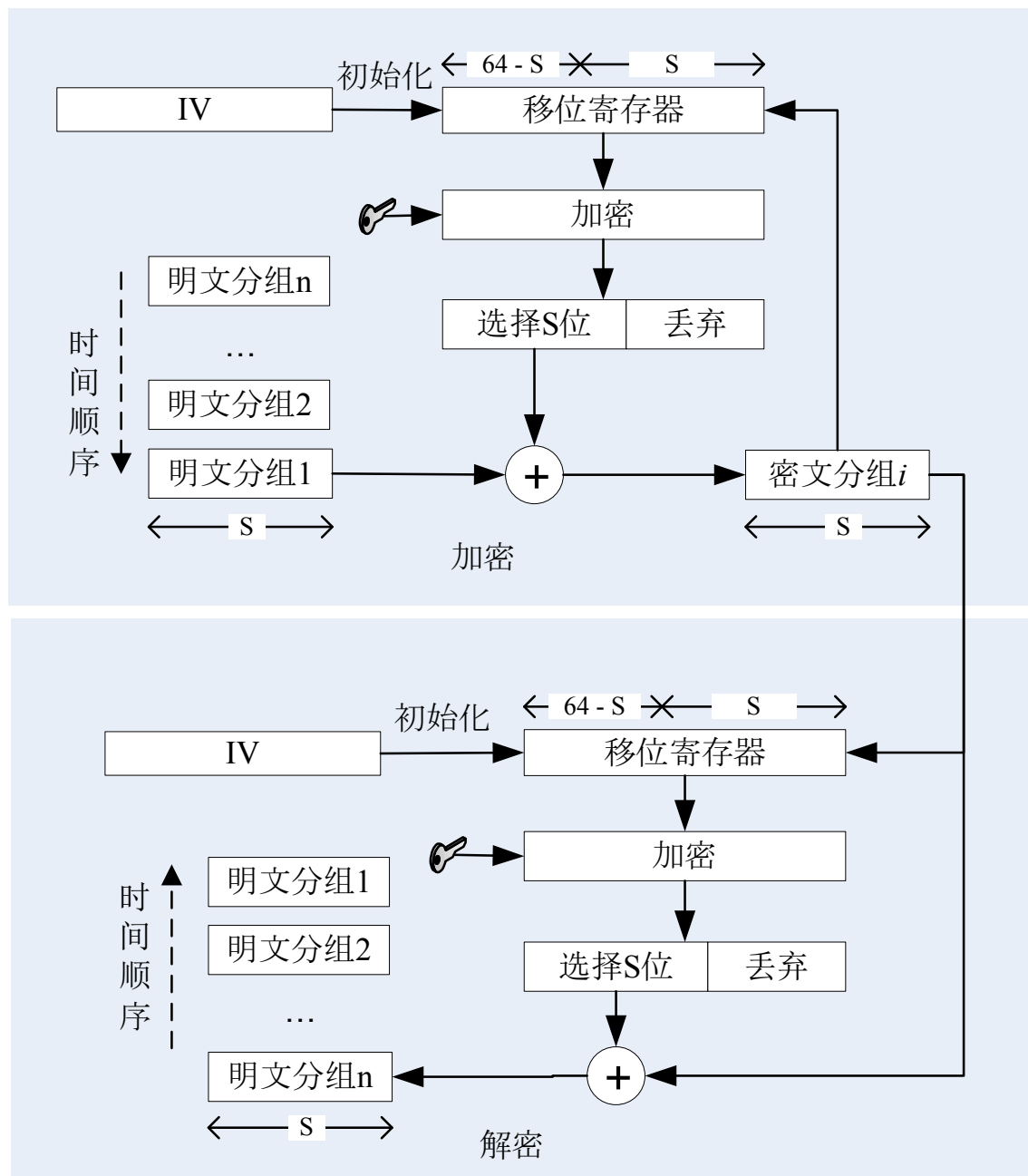


密码分组链接模式（CBC）



CBC工作模式

密码反馈模式 (CFB)



CFB 工作模式

输出反馈模式 (OFB)

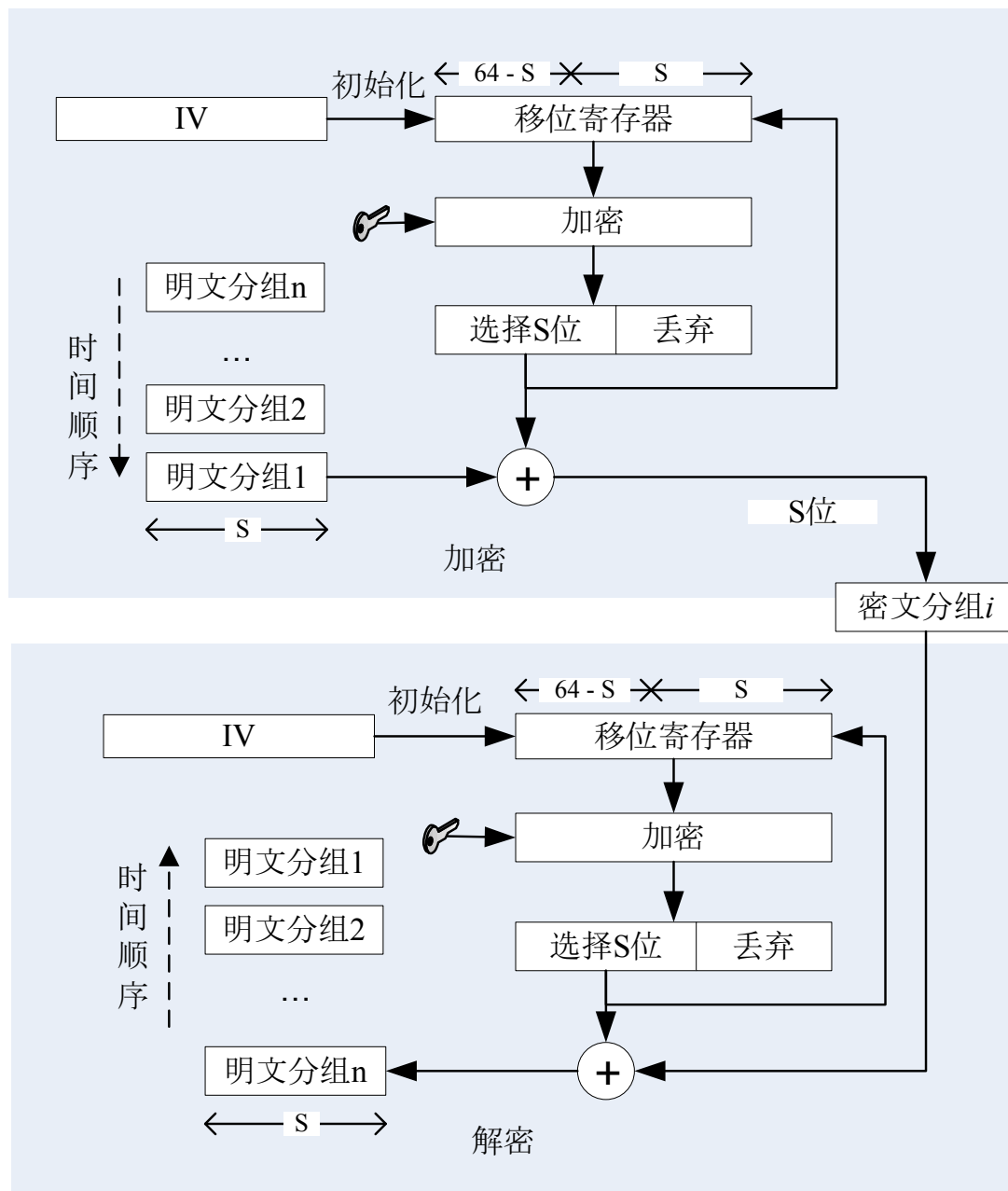


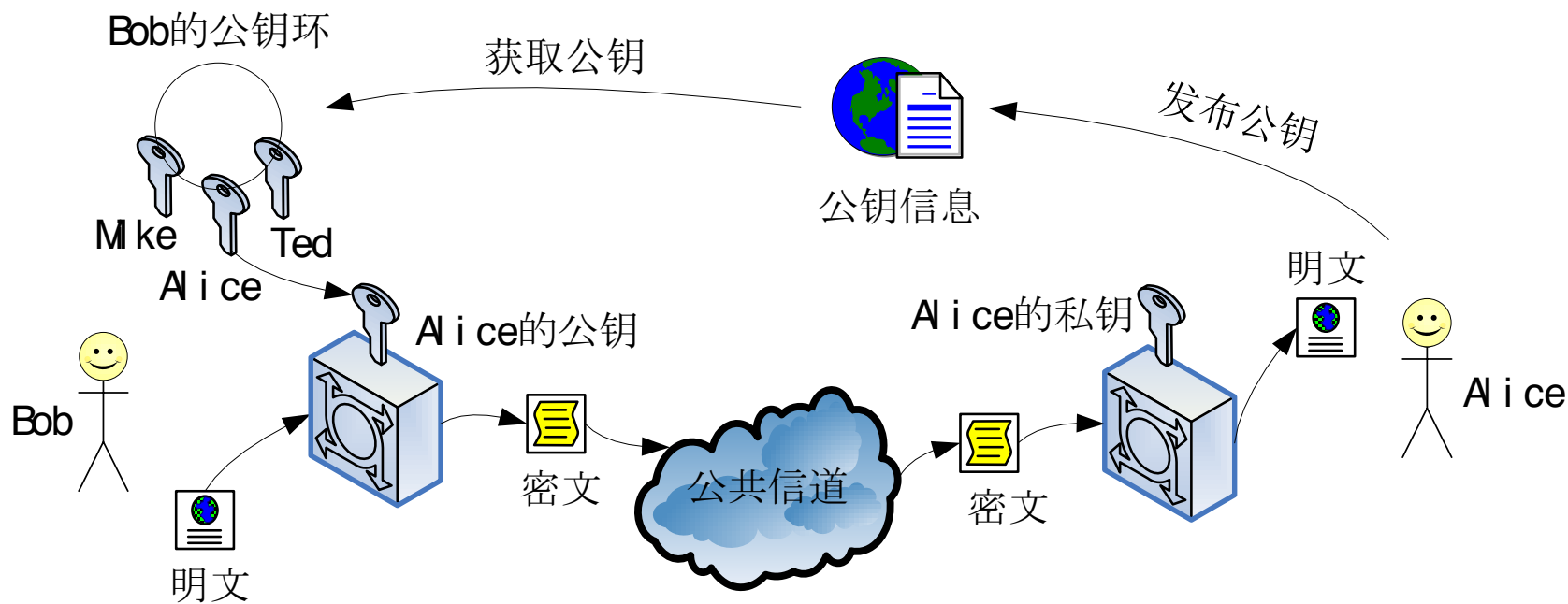
图 2.14 OFB 工作模式

2.3.4其他对称密码简介

- 三重DES
- RC5
- IDEA
- AES算法

2.4 公开密钥密码

- 公开密钥密码又称非对称密钥密码或双密钥密码
 - 加密密钥和解密密钥为两个独立密钥。
 - 公开密钥密码的通信安全性取决于私钥的保密性。



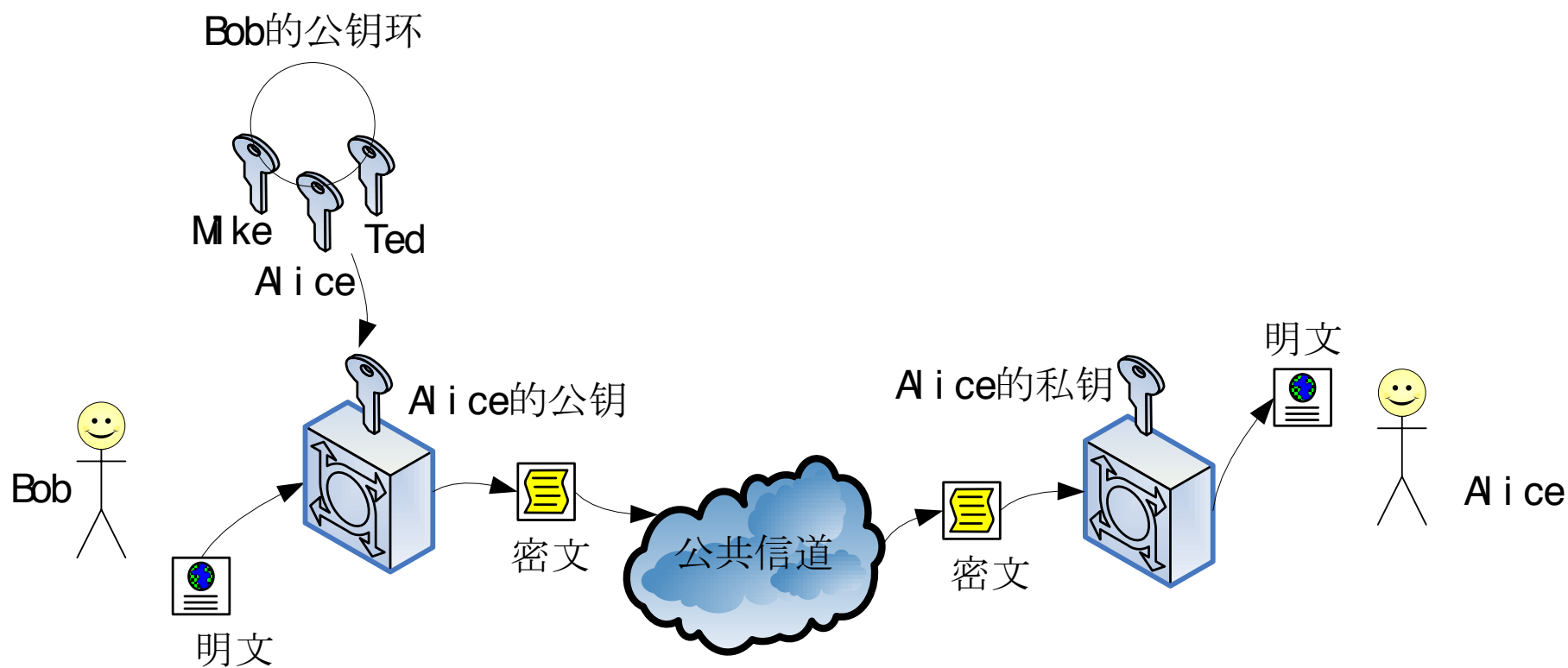
公开密钥密码的模型

2.4.1 公开密钥理论基础

公开密钥密码的核心思想

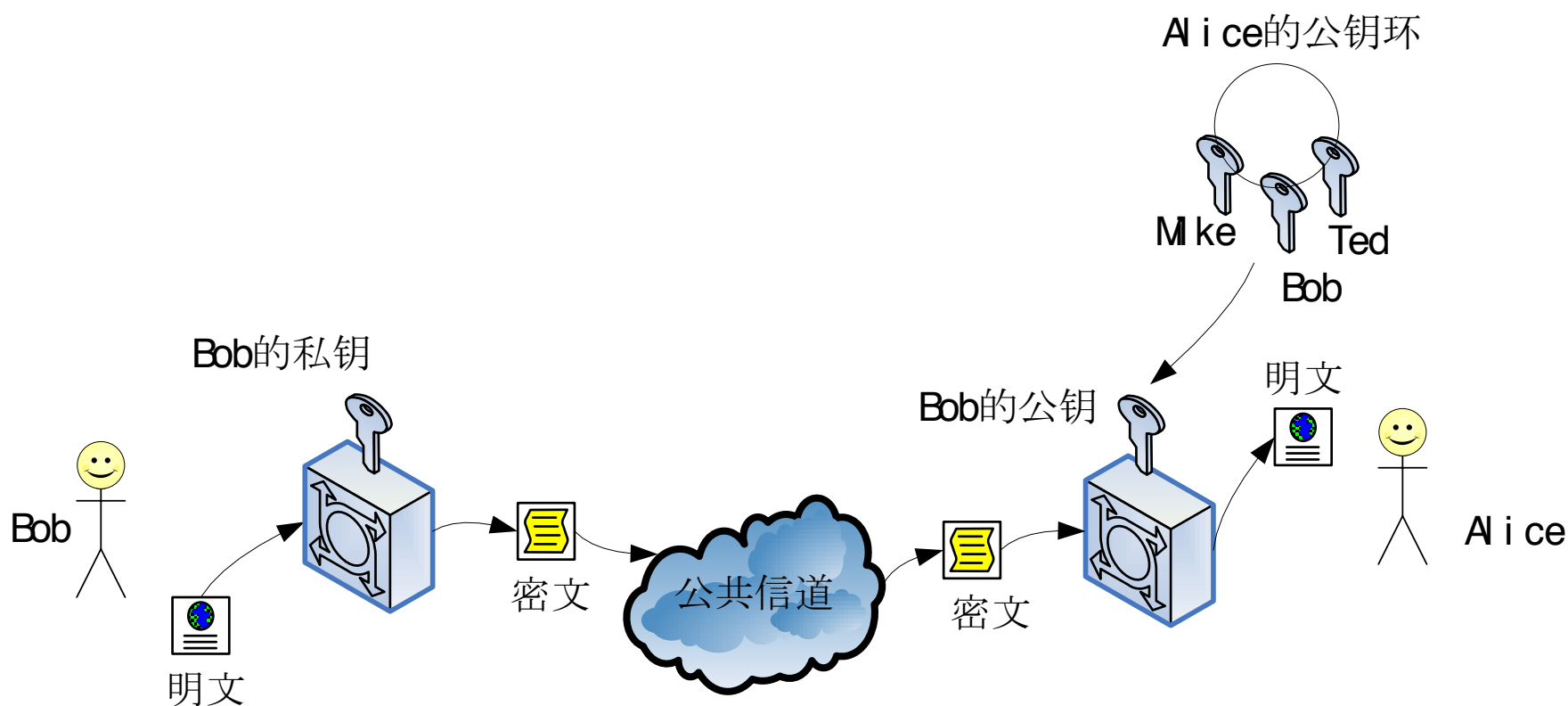
- 公开密钥密码是1976年由Whitfield Diffie和Martin Hellman在其“密码学新方向”一文中提出的。
- 单向陷门函数 $f(x)$ ，必须满足以下三个条件。
 - ① 给定 x ，计算 $y=f(x)$ 是容易的；
 - ② 给定 y ，计算 x 使 $y=f(x)$ 是困难的（所谓计算 $x=f^{-1}(y)$ 困难是指计算上相当复杂已无实际意义）；
 - ③ 存在 δ ，已知 δ 时对给定的任何 y ，若相应的 x 存在，则计算 x 使 $y=f(x)$ 是容易的。

公开密钥的应用：加密模型



公开密钥密码的加密模型

公开密钥的应用：认证模型



公开密钥密码的认证模型

2.4.2 Diffie-Hellman密钥交换算法

- 数学知识

- 原根

- 素数 p 的原根（primitive root）的定义：如果 a 是素数 p 的原根，则数 $a \bmod p, a^2 \bmod p, \dots, a^{p-1} \bmod p$ 是不同的并且包含从1到 $p-1$ 之间的所有整数的某种排列。对任意的整数 b ，可以找到唯一的幂 i ，满足 $b \equiv a^i \bmod p$ ，且 $1 \leq i \leq p-1$ 。

- 注：“ $b \equiv a \bmod p$ ”等价于“ $b \bmod p = a \bmod p$ ”，称为“ b 与 a 模 p 同余”。

— 离散对数

- 若 a 是素数 p 的一个原根，则相对于任意整数 b ($b \bmod p \neq 0$)，必然存在唯一的整数 i ($1 \leq i \leq p-1$)，使得 $b \equiv a^i \bmod p$ ， i 称为 b 的以 a 为基数且模 p 的幂指数，即离散对数。
- 对于函数 $y \equiv g^x \bmod p$ ，其中， g 为素数 p 的原根， y 与 x 均为正整数，已知 g 、 x 、 p ，计算 y 是容易的；而已知 y 、 g 、 p ，计算 x 是困难的，即求解 y 的离散对数 x 。
- 注：离散对数的求解为数学界公认的困难问题。

Diffie-Hellman密钥交换算法

- Alice和Bob协商好一个大素数 p ，和大的整数 g ， $1 < g < p$ ， g 是 p 的原根。 p 和 g 无须保密，可为网络上的所有用户共享。当Alice和Bob要进行保密通信时，他们可以按如下步骤来做：
 - ① Alice选取大的随机数 $x < p$ ，并计算 $Y = g^x \pmod{P}$;
 - ② Bob选取大的随机数 $x' < p$ ，并计算 $Y' = g^{x'} \pmod{P}$;
 - ③ Alice将 Y 传送给Bob，Bob将 Y' 传送给Alice;
 - ④ Alice计算 $K = (Y')^x \pmod{P}$ ，Bob计算 $K' = (Y)^{x'} \pmod{P}$
- 显而易见 $K = K' = g^{xx'} \pmod{P}$ ，即Alice和Bob已获得了相同的秘密值 K 。

2.4.3 RSA公开密钥算法

- 欧拉定理

- 欧拉函数是欧拉定理的核心概念，其表述：对于一个正整数 n ，由小于 n 且和 n 互素的正整数构成的集合为 Z_n ，这个集合被称为 n 的完全余数集合。 Z_n 包含的元素个数记做 $\phi(n)$ ，称为欧拉函数，其中 $\phi(1)$ 被定义为1，但是并没有任何实质的意义。
- 如果两个素数 p 和 q ，且 $n = p \times q$ ，则 $\phi(n) = (p-1)(q-1)$ ；
- 欧拉定理的具体表述：正整数 a 与 n 互素，则 $a^{\phi(n)} \equiv 1 \pmod n$ 。

- 推论：

- 给定两个素数 p 和 q ，以及两个整数 m 、 n ，使得 $n = p \times q$ ，且 $0 < m < n$ ，对于任意整数 k 下列关系成立， $m^{k\phi(n)+1} = m^{k(p-1)(q-1)+1} \equiv m \pmod n$ 。

大整数因子分解

- 大整数因子分解问题：

- 已知 p 、 q 为两个大素数，则求 $N=p \times q$ 是容易的，只需要一次乘法运算；但已知 N 是两个大素数的乘积，要求将 N 分解，则在计算上是困难的，其运行时间复杂程度接近于不可行。

- 算法时间复杂性：

- 如果输入规模为 n 时，一个算法的运行时间复杂度为 $O(n)$ ，称此算法为线性的；
- 运行时间复杂度为 $O(n^k)$ ，其中 k 为常量，称此算法为多项式的；
- 若有某常量 t 和多项式 $h(n)$ ，使算法的运行时间复杂度为 $O(t^{h(n)})$ ，则称此算法为指数的。

- 一般说来，

- 在线性时间和多项式时间内被认为是可解决的，比多项式时间更坏的，尤其是指数时间被认为是不可解决的。
- 注：如果输入规模太小，即使很复杂的算法也会变得可行的。

RSA密码算法

- RSA密码体制：
 - 明文和密文均是0到 n 之间的整数， n 通常为1024位二进制数或309位十进制数，
 - 明文空间 P =密文空间 $C=\{x \in \mathbb{Z} \mid 0 < x < n, \mathbb{Z} \text{ 为整数集合}\}$ 。
- RSA密码的密钥生成具体步骤如下：
 - ① 选择互异的素数 p 和 q ，计算 $n=pq$ ， $\varphi(n) = (p-1)(q-1)$ ；
 - ② 选择整数 e ，使 $\gcd(\varphi(n), e) = 1$ ，且 $1 < e < \varphi(n)$ ；
 - ③ 计算 d ， $d \equiv e^{-1} \bmod \varphi(n)$ ，即 d 为模 $\varphi(n)$ 下 e 的乘法逆元；
- 公钥 $P_k = \{ e, n \}$ ，私钥 $S_k = \{ d, n, p, q \}$
- 加密： $c = m^e \bmod n$ ；解密： $m = c^d \bmod n$ 。

RSA例

- $p=101$, $q=113$, $n=11413$, $\phi(n)=100 \times 112 = 11200$ 。
- $e = 3533$, 求得 $d \equiv e^{-1} \bmod 11200 \equiv 6597 \bmod 11200$, $d = 6597$ 。
- 公开 $n=11413$ 和 $e=3533$,
- 明文 9726, 计算 $9726^{3533} \bmod 11413 = 5761$, 发送密文 5761。
- 密文 5761 时, 用 $d=6597$ 进行解密, 计算 $5761^{6597} \bmod 11413 = 9726$ 。

RSA的安全性

- RSA是基于单向函数 $e_k(x)=x^e \pmod n$ ，求逆计算不可行。
- 解密的关键是了解陷门信息，即能够分解 $n=pq$ ，知道 $\phi(n)=(p-1)(q-1)$ ，从而解出解密私钥 d 。
- 如果要求RSA是安全的， p 与 q 必为足够大的素数;使分析者没有办法在多项式时间内将 n 分解出来。
- 模 n 的求幂运算
 - 著名的“平方-和-乘法”方法将计算 $x^c \pmod n$ 的模乘法的次数缩小到至多为 $2l$ ， l 是指数 c 二进制表示的位数。

2.4.4 其他公开密钥密码简介

- 基于大整数因子分解问题：
 - RSA密码、Rabin密码
- 基于有限域上的离散对数问题：
 - Differ-Hellman公钥交换体制、ElGamal密码
- 基于椭圆曲线上的离散对数问题：
 - Differ-Hellman公钥交换体制、ElGamal密码。

2.5 消息认证

- 2.5.1 概述

- 威胁信息完整性的行为主要包括：

- 伪造：假冒他人的信息源向网络中发布消息；
 - 内容修改：对消息的内容进行插入、删除、变换和修改；
 - 顺序修改：对消息进行插入、删除或重组消息序列；
 - 时间修改：针对网络中的消息，实施延迟或重放；
 - 否认：接受者否认收到消息，发送者否认发送过消息。

- 消息认证是保证信息完整性的重要措施

- 其目的主要包括：

- 证明：消息的信源和信宿的真实性；
 - 证明：消息内容是否曾受到偶然或有意的篡改，
 - 证明：消息的序号和时间性是否正确。

— 消息认证由具有认证功能的函数来实现的

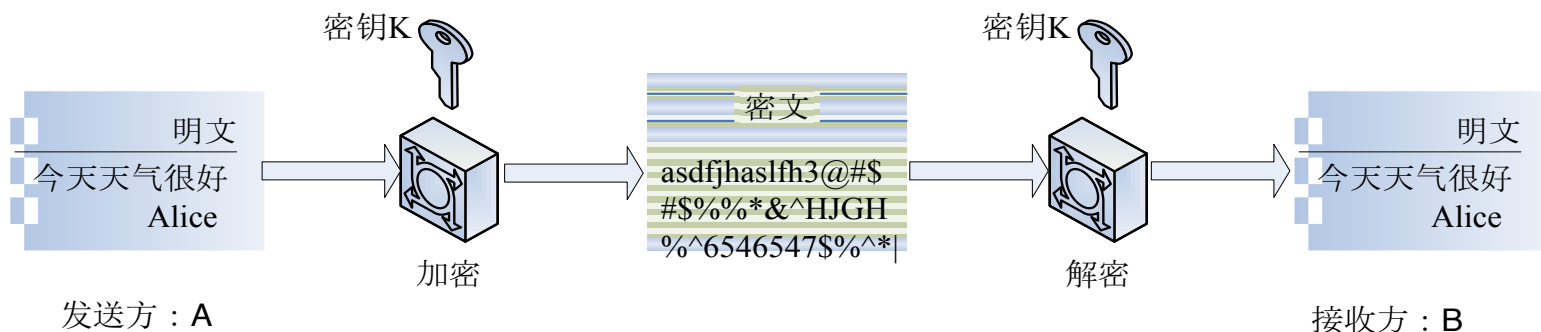
- **消息加密**，用消息的完整密文作为消息的认证符；
- **消息认证码MAC**（Message Authentication Code），也称密码校验和，使用密码对消息加密，生成固定长度的认证符；
- **消息编码**，是针对信源消息的编码函数，使用编码抵抗针对消息的攻击。

2.5.2 认证函数

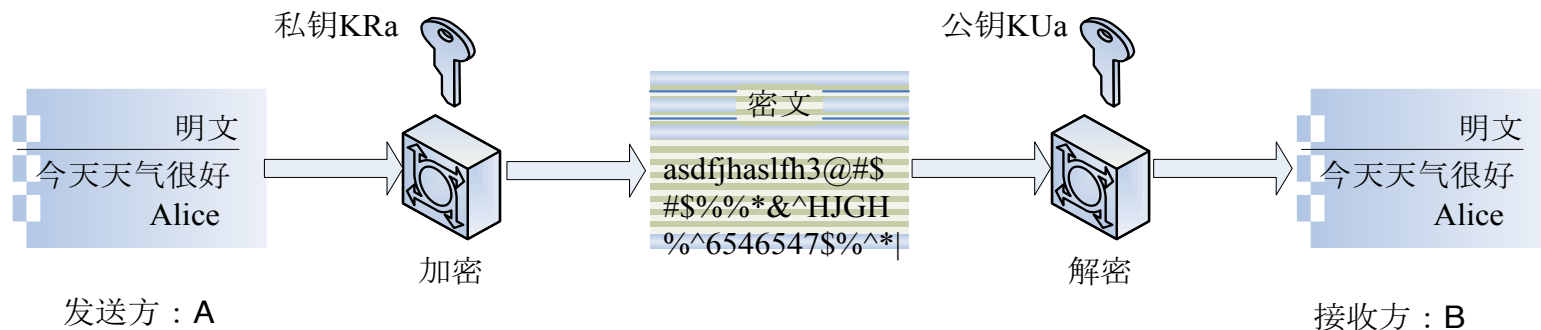
- 认证技术在功能上可以分为两层
 - 下层包含一个产生认证符的函数，认证符是一个用来认证消息的值；
 - 上层是以认证函数为原语，接收方可以通过认证函数来验证消息的真伪。

一、消息加密函数

- 对称密钥密码对消息加密，不仅具有机密性，同时也具有一定的可认证性；
- 公开密钥密码本身就提供认证功能，其具有的私钥加密、公钥解密以及反之亦然特性；



(a) 对称密钥密码：加密和认证



(b) 公开密钥密码：认证

二、消息认证码

- 消息认证码MAC的基本思想：
 - 利用事先约定的密码，加密生成一个固定长度的短数据块MAC，并将MAC附加到消息之后，一起发送给接收者；
 - 接收者使用相同密码对消息原文进行加密得到新的MAC，比较新的MAC和随消息一同发来的MAC，如果相同则未受到篡改。

- 生成消息认证码的方法：
 - 基于加密函数的认证码和消息摘要（在散列函数中讨论）。
 - 消息认证符可以是整个64位的 O_n ，也可以是 O_n 最左边的M位

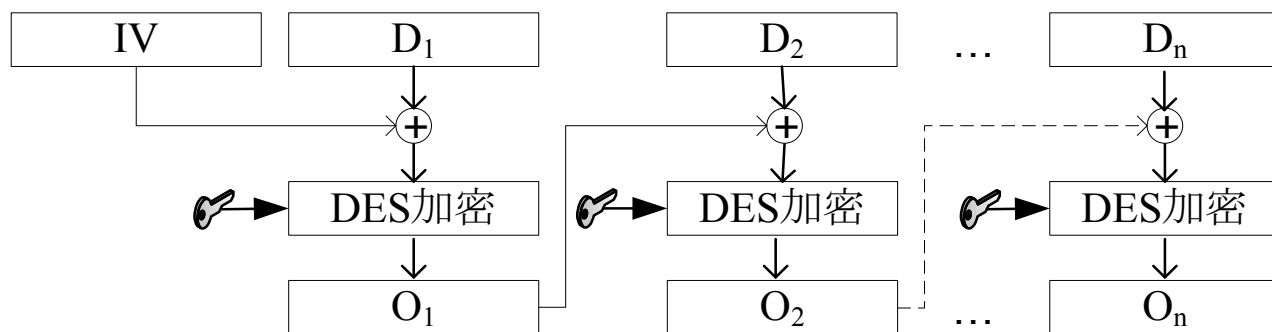


图2.19 基于DES的消息认证码

三、消息编码

- 消息编码认证的基本思想：
 - 引入冗余度，使通过信道传送的可能序列集 M （编码集）大于消息集 S （信源集）。
 - 发送方从 M 中选出用来代表消息的许用序列 L_i ，即对信息进行编码；
 - 接收方根据编码规则，进行解码，还原出发送方按此规则向他传来的消息。
 - 窜扰者不知道被选定的编码规则，因而所伪造的假码字多是 M 中的禁用序列，接收方将以很高的概率将其检测出来，并拒绝通过认证。

- 如果决定采用 L_0 ，则以发送消息“00”代表信源“0”，发送消息“10”代表信源“1”。在子规则 L_0 下，消息“00”和“10”是合法的，而消息“01”和“11”在 L_0 之下不合法，收方将拒收这两个消息。

信源S 编码 法则L	0	1	禁用序列
L_0	00	10	01, 11
L_1	00	11	01, 10
L_2	01	10	00, 11
L_3	01	11	00, 10

2.5.3 散列函数

- 散列函数（Hash Function）的目的
 - 将任意长的消息映射成一个固定长度的散列值（hash值），也称为消息摘要。消息摘要可以作为认证符，完成消息认证。
- 散列函数的健壮性
 - 弱无碰撞特性
 - 散列函数 h 被称为是弱无碰撞的，是指在消息特定的明文空间 X 中，给定消息 $x \in X$ ，在计算上几乎找不到不同于 x 的 x' ， $x' \in X$ ，使得 $h(x)=h(x')$ 。
 - 强无碰撞特性
 - 散列函数 h 被称为是强无碰撞的，是指在计算上难以找到与 x 相异的 x' ，满足 $h(x)=h(x')$ ， x' 可以不属于 X 。
 - 单向性
 - 散列函数 h 被称为单向的，是指通过 h 的逆函数 h^{-1} 来求得散列值 $h(x)$ 的消息原文 x ，在计算上不可行。

散列值的安全长度

— “生日悖论”

- 如果一个房间里有23个或23个以上的人，那么至少有两个人的生日相同的概率要大于50%。对于60或者更多的人，这种概率要大于99%。
- 不计特殊的闰年，计算房间里所有人的生日都不相同的概率，

①第一个人不发生生日冲突的概率是 $\frac{365}{365}$,

②第二个人不发生生日冲突的概率是 $1 - \frac{1}{365}$, ...,

③第n个人是 $1 - \frac{n-1}{365}$,

④所有人生日都不冲突的概率是：

$$\text{① } E = 1 \times \left(1 - \frac{1}{365}\right) \times \dots \times \left(1 - \frac{n-2}{365}\right) \times \left(1 - \frac{n-1}{365}\right),$$

⑤而发生冲突的概率 $P = 1 - E$ ，当 $n = 23$ 时， $P \approx 0.507$ ；
 $n = 100$ 时， $P \approx 0.99999996$ 。

散列值的安全长度

— 生日悖论对于散列函数的意义

- n 位长度的散列值，可能发生一次碰撞的测试次数不是 2^n 次，而是大约 $2^{n/2}$ 次。
- 一个40位的散列值将是不安全的，因为大约100万个随机散列值中将找到一个碰撞的概率为50%，
- 消息摘要的长度不低于为128位。

MD5

- 1991年Rivest对MD4的进行改进升级，提出了MD5（Message Digest Algorithm 5）。
- MD5具有更高的安全性，目前被广泛使用。

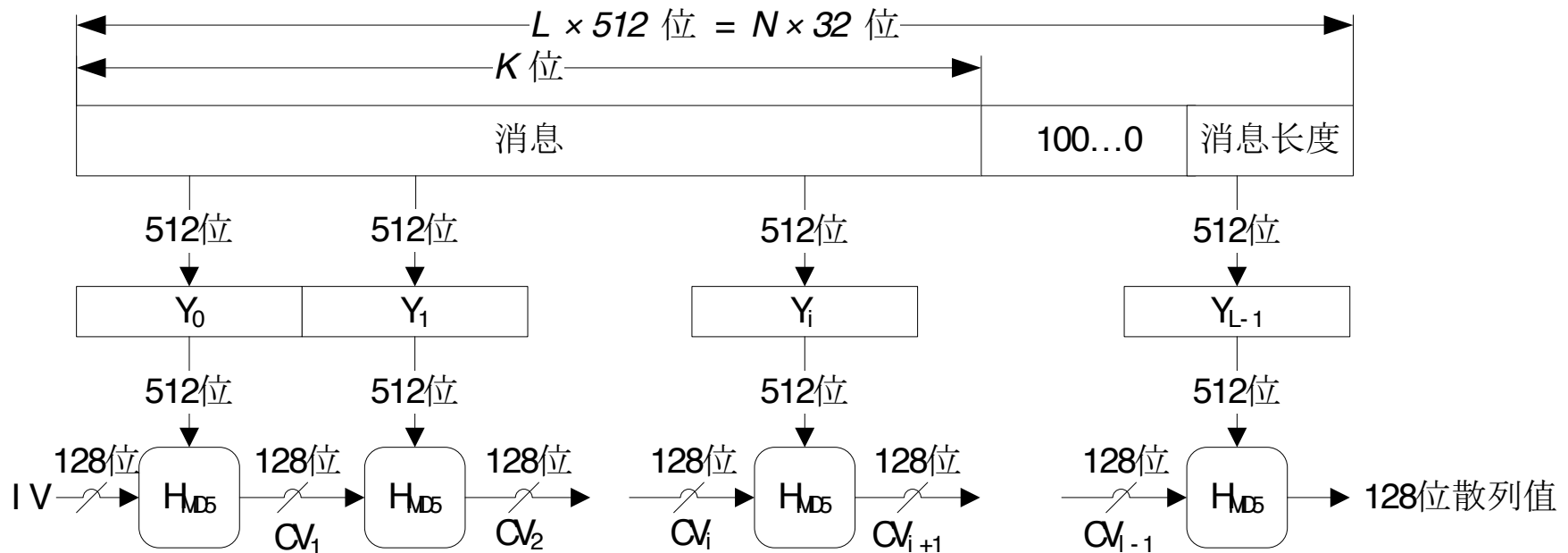


图2. 20 MD5算法

MD5

- 四轮运算涉及四个函数：

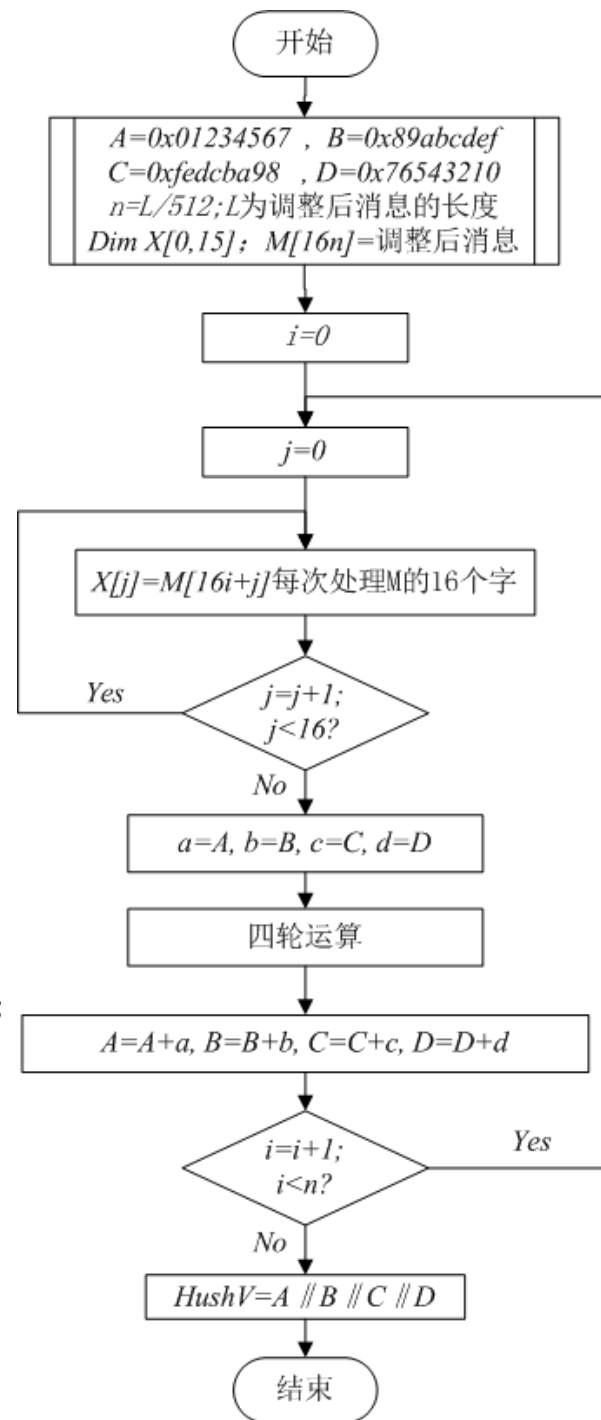
- $E(X, Y, Z) = (X \wedge Y) \vee ((\neg X) \wedge Z)$
- $F(X, Y, Z) = (X \wedge Z) \vee (Y \wedge (\neg Z))$
- $G(X, Y, Z) = X \oplus Y \oplus Z$
- $H(X, Y, Z) = Y \oplus (X \vee (\neg Z))$

- 四轮运算：

- $EE(a,b,c,d,M_j,s,t_i): a = b + ((a+(E(b,c,d)+M_j+t_i)<<s);$
- $FF(a,b,c,d,M_j,s,t_i): a = b + ((a+(F(b,c,d)+ M_j + t_i)<<s);$
- $GG(a,b,c,d,M_j,s,t_i) : a = b + ((a+(G(b,c,d)+ M_j + t_i)<<s);$
- $HH(a,b,c,d, M_j,s,t_i) : a = b + ((a+(H(b,c,d)+ M_j + t_i)<<s);$

- 得到常数 t_i 的计算方法是：

- 整个四轮操作总共分为64步，在第 i 步中 t_i 是 $2^{32} \times \text{abs}(\sin(i))$ 的整数部分， i 的单位是弧度



第一轮

- $EE(a,b,c,d,M_j,s,t_i): a = b + ((a+(E(b,c,d)+M_j+t_i)<<s);$

- (1) $EE(a,b,c,d,M_0,7,0xd76aa478)$ — $\gg a = b + ((a+(E(b,c,d)+M_0+0xd76aa478)<<7);$
- (2) $EE(d,a,b,c,M_1,12,0xe8c7b756)$
- (3) $EE(c,d,a,b,M_2,17,0x242070db)$
- (4) $EE(b,c,d,a,M_3,22,0xc1bdceee)$
- (5) $EE(a,b,c,d,M_4,7,0xf57c0faf)$
- (6) $EE(d,a,b,c,M_5,12,0x4787c62a)$
- (7) $EE(c,d,a,b,M_6,17,0xa8304613)$
- (8) $EE(b,c,d,a,M_7,22,0xfd469501)$
- (9) $EE(a,b,c,d,M_8,7,0x698098d8)$
- (10) $EE(d,a,b,c,M_9,12,0x8b44f7af)$
- (11) $EE(c,d,a,b,M_{10},17,0xffff5bb1)$
- (12) $EE(b,c,d,a,M_{11},22,0x895cd7be)$
- (13) $EE(a,b,c,d,M_{12},7,0x6b901122)$
- (14) $EE(d,a,b,c,M_{13},12,0xfd987193)$
- (15) $EE(c,d,a,b,M_{14},17,0xa679438e)$
- (16) $EE(b,c,d,a,M_{15},22,0x49b40821)$

第二轮

- $FF(a,b,c,d,M_j,s,t_i): a = b + ((a+(F(b,c,d)+ M_j + t_i) \ll s)$

- (1) $FF(a,b,c,d,M_1,5,0xf61e2562)$ — $\gg a = b + ((a+(F(b,c,d)+M_1+0xf61e2562) \ll 5);$
(2) $FF(d,a,b,c,M_6,9,0xc040b340)$
(3) $FF(c,d,a,b,M_{11},14,0x265e5a51)$
(4) $FF(b,c,d,a,M_0,20,0xe9b6c7aa)$
(5) $FF(a,b,c,d,M_5,5,0xd62f105d)$
(6) $FF(d,a,b,c,M_{10},9,0x02441453)$
(7) $FF(c,d,a,b,M_{15},14,0xd8a1e681)$
(8) $FF(b,c,d,a,M_4,20,0xe7d3fbc8)$
(9) $FF(a,b,c,d,M_9,5,0x21e1cde6)$
(10) $FF(d,a,b,c,M_{14},9,0xc33707d6)$
(11) $FF(c,d,a,b,M_3,14,0xf4d50d87)$
(12) $FF(b,c,d,a,M_8,20,0x455a14ed)$
(13) $FF(a,b,c,d,M_{13},5,0xa9e3e905)$
(14) $FF(d,a,b,c,M_2,9,0xfcefa3f8)$
(15) $FF(c,d,a,b,M_7,14,0x676f02d9)$
(16) $FF(b,c,d,a,M_{12},20,0x8d2a4c8a)$)

第三轮

• $GG(a,b,c,d,M_j,s,t_i) : a = b + ((a+(G(b,c,d)+ M_j + t_i) \ll s);$

(1) $GG(a,b,c,d,M_5,4,0xEEfa3942)$ — $\gg a = b + ((a+(G(b,c,d)+M_5+0xEEfa3942) \ll 4);$

(2) $GG(d,a,b,c,M_8,11,0x8771f681)$

(3) $GG(c,d,a,b,M_{11},16,0x6d9d6122)$

(4) $GG(b,c,d,a,M_{14},23,0xfde5380c)$

(5) $GG(a,b,c,d,M_1,4,0xa4beea44)$

(6) $GG(d,a,b,c,M_4,11,0x4bdecfa9)$

(7) $GG(c,d,a,b,M_7,16,0xf6bb4b60)$

(8) $GG(b,c,d,a,M_{10},23,0xbefbfc70)$

(9) $GG(a,b,c,d,M_{13},4,0x289b7ec6)$

(10) $GG(d,a,b,c,M_0,11,0xea127fa)$

(11) $GG(c,d,a,b,M_3,16,0xd4ef3085)$

(12) $GG(b,c,d,a,M_6,23,0x04881d05)$

(13) $GG(a,b,c,d,M_9,4,0xd9d4d039)$

(14) $GG(d,a,b,c,M_{12},11,0xe6db99e5)$

(15) $GG(c,d,a,b,M_{15},16,0x1fa27cf8)$

(16) $GG(b,c,d,a,M_2,23,0xc4ac5665)$

第四轮

- $HH(a,b,c,d, M_j, s, t_i) : a = b + ((a+(H(b,c,d)+ M_j + t_i) \ll s);$

- (1) $HH(a,b,c,d,M_0,6,0xf4292244)$ — $\gg a = b + ((a+(H(b,c,d)+M_0+0xf4292244) \ll 6);$
- (2) $HH(d,a,b,c,M_7,10,0x432aEE97)$
- (3) $HH(c,d,a,b,M_{14},15,0xab9423a7)$
- (4) $HH(b,c,d,a,M_5,21,0xfc93a039)$
- (5) $HH(a,b,c,d,M_{12},6,0x655b59c3)$
- (6) $HH(d,a,b,c,M_3,10,0x8f0ccc92)$
- (7) $HH(c,d,a,b,M_{10},15,0xEEeEE47d)$
- (8) $HH(b,c,d,a,M_1,21,0x85845dd1)$
- (9) $HH(a,b,c,d,M_8,6,0x6fa87e4f)$
- (10) $HH(d,a,b,c,M_{15},10,0xfe2ce6e0)$
- (11) $HH(c,d,a,b,M_6,15,0xa3014314)$
- (12) $HH(b,c,d,a,M_{13},21,0x4e0811a1)$
- (13) $HH(a,b,c,d,M_4,6,0xf7537e82)$
- (14) $HH(d,a,b,c,M_{11},10,0xbd3af235)$
- (15) $HH(c,d,a,b,M_2,15,0x2ad7d2bb)$
- (16) $HH(b,c,d,a,M_9,21,0xeb86d391)$

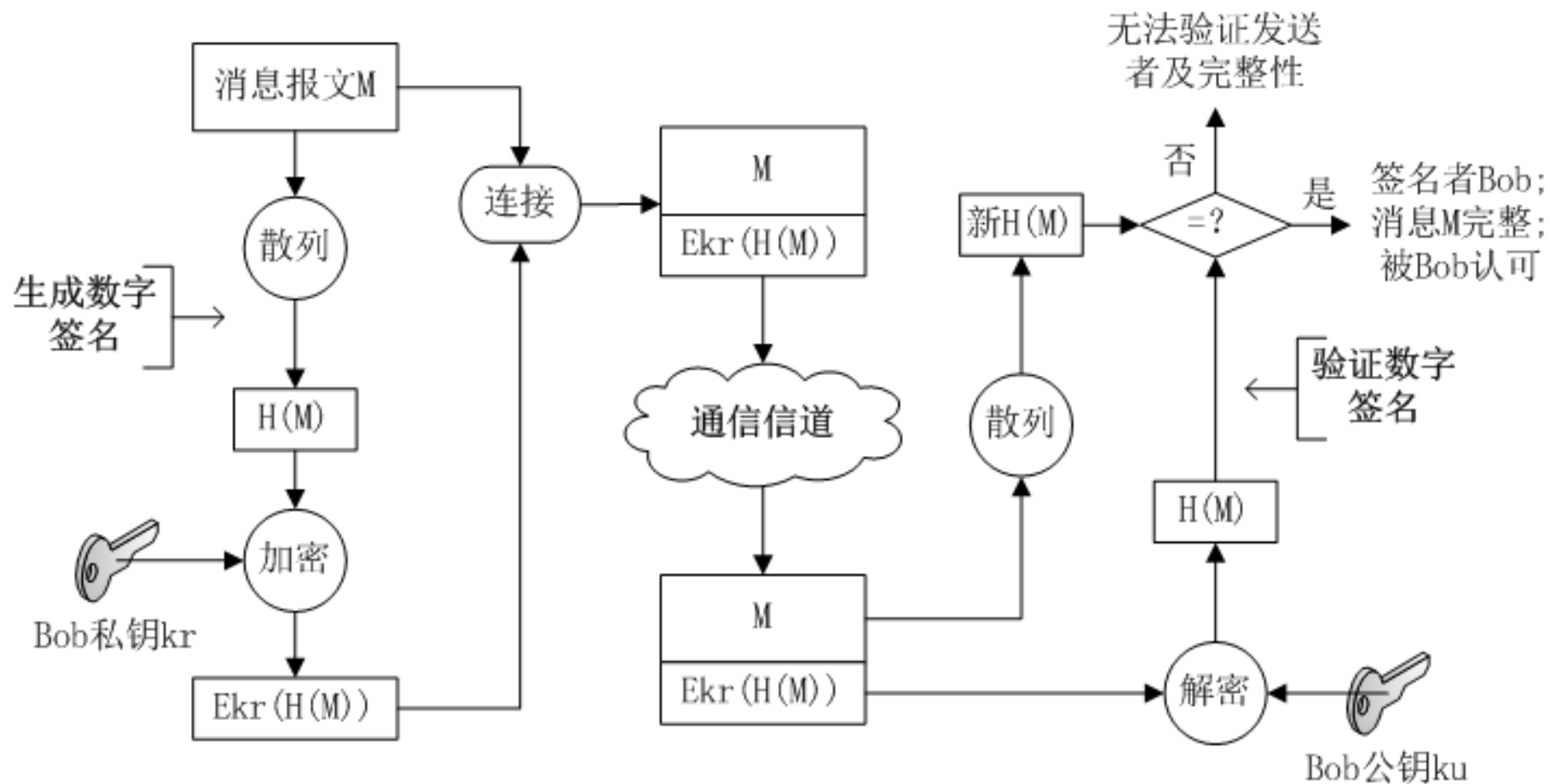
2.5.4 数字签名

- 数字签名：Digital Signature，
- 在ISO7498-2标准定义为
 - “附加在数据单元上的一些数据或是对数据单元所作的密码变换，这种数据或变换可以被数据单元的接收者用来确认数据单元来源和数据单元的完整性，并保护数据不会被人（例如接收者）伪造”。
- 美国电子签名标准对数字签名作了如下解释：
 - “数字签名是利用一套规则和一个参数对数据进行计算所得的结果，用此结果能够确认签名者的身份和数据的完整性”
- 一般来说，数字签名可以被理解为：
 - 通过某种密码运算生成一系列符号及代码，构成可以用来进行数据来源验证的数字信息。

- 从签名形式上分，数字签名有两种
 - 一种是对整个消息的签名，
 - 一种是对压缩消息的签名，
 - 它们都是附加在被签名消息之后或在某一特定位置上的一段数据信息。
- 数字签名主要目的
 - 保证收方能够确认或验证发方的签名，但不能伪造；发方发出签名消息后，不能否认所签发的消息。

- 设计数字签名必须满足下列条件：
 - 签名必须基于一个待签名信息的位串模板；
 - 签名必须使用某些对发送方来说是唯一的信息，以防止双方的伪造与否认；
 - 必须相对容易生成、识别和验证数字签名；
 - 伪造该数字签名在计算复杂性意义上具有不可行性
 - 既包括对一个已有的数字签名构造新的消息，也包括对一个给定消息伪造一个数字签名。

数字签名的生成及验证



2.6 密码学新进展

- 1989年，英国数学家Matthews，基于混沌的加密技术混沌密码学
 - 混沌系统具有良好的伪随机特性、轨道的不可预测性、对初始状态及控制参数的敏感性等一系列特性，
 - 传统的密码算法敏感性依赖于密钥，而混沌映射依赖于初始条件和映射中的参数；
 - 传统的加密算法通过加密轮次来达到扰乱和扩散，混沌映射则通过迭代，将初始域扩散到整个相空间；
 - 传统加密算法定义在有限集上，而混沌映射定义在实数域内。

- 量子密码

- 1970年威斯纳提出利用单量子态制造不可伪造的“电子钞票”，这个构想由于量子态的寿命太短而无法实现，
- 1984年，IBM的贝内特和加拿大学者布拉萨德提出了第一个量子密码方案，由此迎来了量子密码学的新时期。
- 量子密码体系采用量子态作为信息载体，经由量子通道在合法的用户之间传送密钥。
- 量子密码的安全性由量子力学原理所保证，被称为是绝对安全的。
- 所谓绝对安全是指即使在窃听者可能拥有极高的智商、可能采用最高明的窃听措施、可能使用最先进的测量手段，密钥的传送仍然是安全的，可见量子密码研究具有极其重大的意义。

- DNA计算

- 1994年，Adleman等科学家进行了世界上首次DNA计算，解决了一个7节点有向汉密尔顿回路问题。
- 由于DNA计算具有的信息处理的高并行性、超高容量的存储密度和超低的能量消耗等特点，非常适合用于攻击密码计算系统的不同部分，对传统的基于计算安全的密码体制提出了挑战。

Any question?