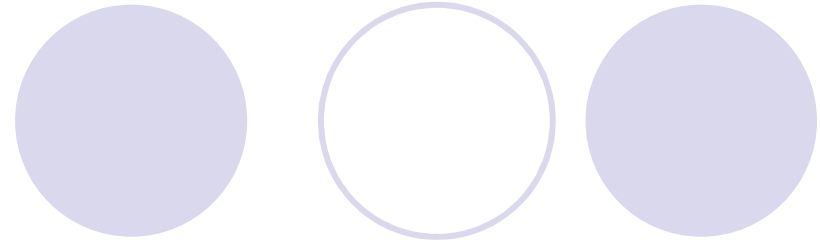


2.3 对称密钥密码

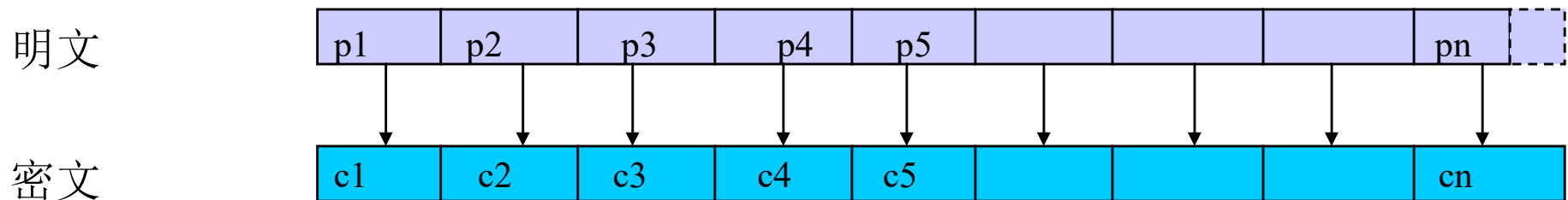
- 分组密码
- 数据加密标准



对称密码算法类型

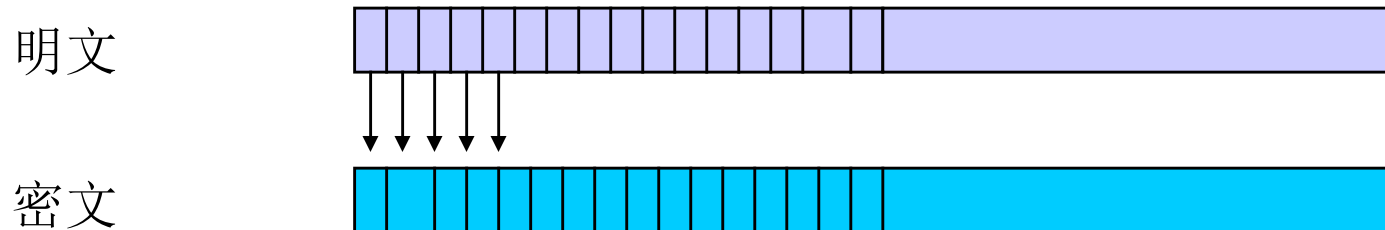
- 分组密码

在明文分组和密文分组上进行运算——通常分组长大于或等于64bits。相同的明文和相同的密钥得到相同的密文。



- 流密码

作用在明文和密文的数据序列的1 bit 或1 byte 上。



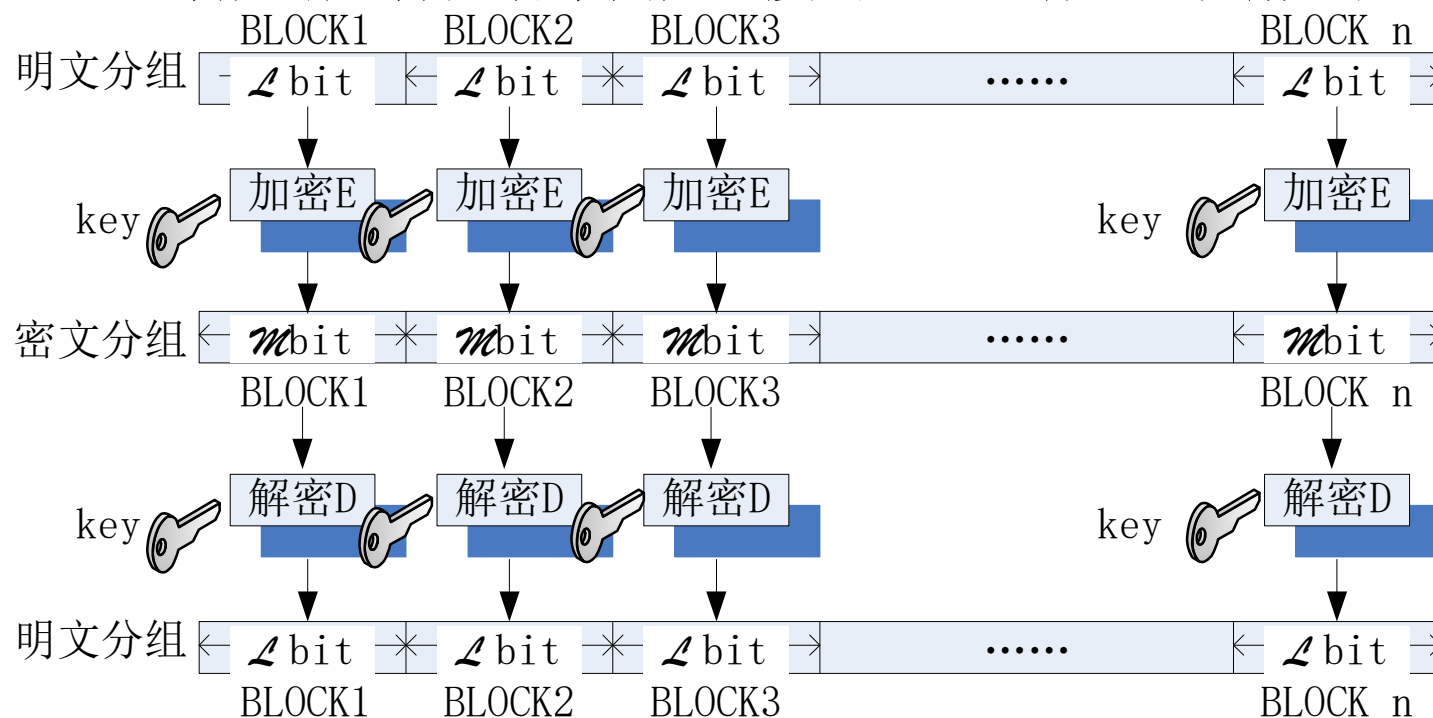
对称密钥密码加密模式

- 对称密码加密系统从工作方式上可分为：

- 分组密码、序列密码

- 分组密码原理：

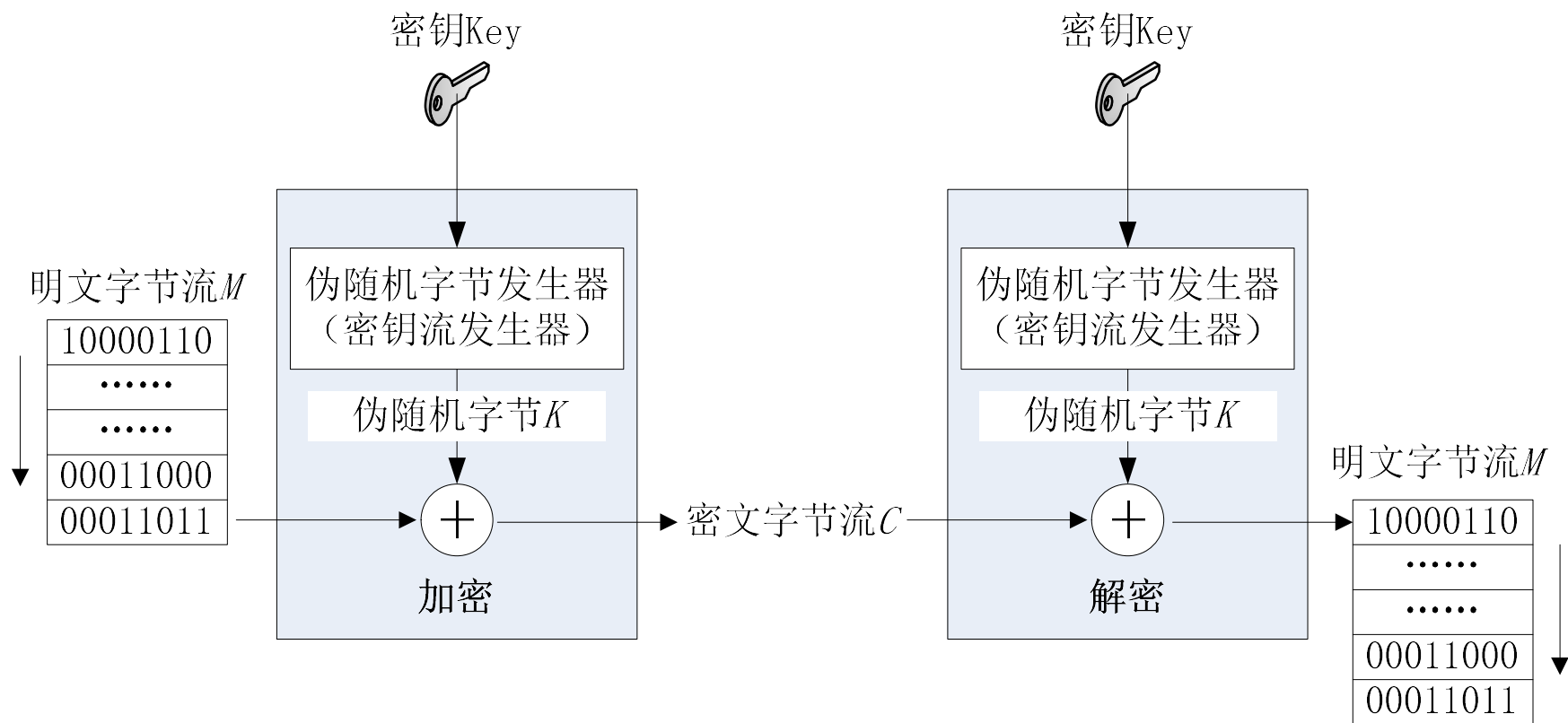
- 明文消息分成若干固定长度的组，进行加密；解密亦然。



分组密码工作原理示意图

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。

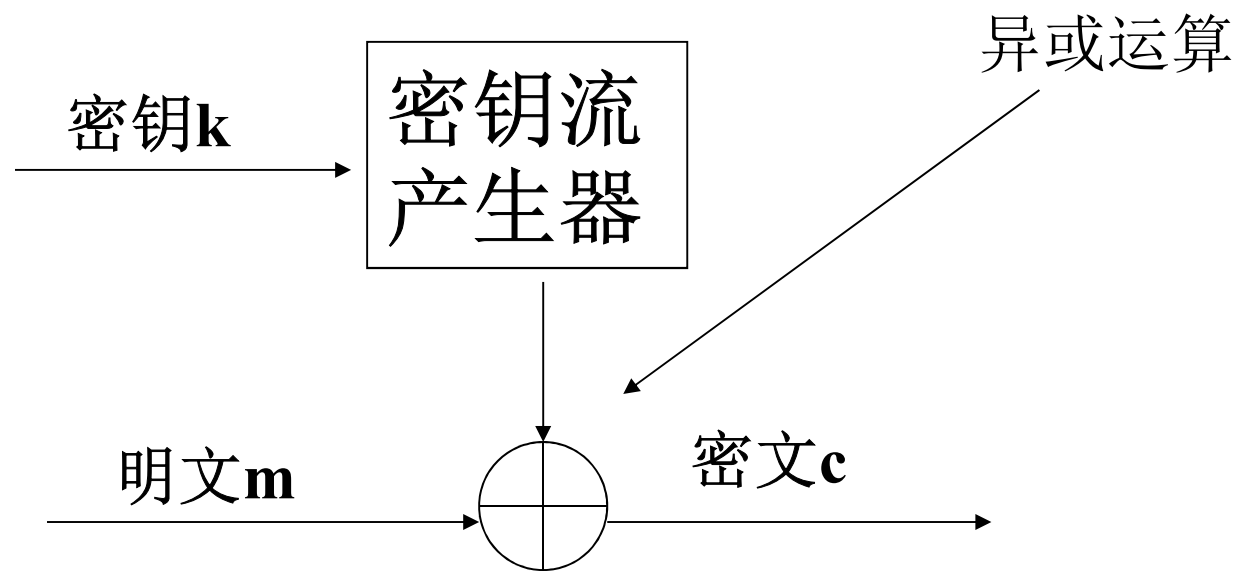


序列密码工作原理示意图

流密码模型

流密码

每次加密数据流的一位或一个字节



流密码体制模型

分组密码



分组密码

将一个明文组作为整体加密且通常得到的是与之等长的密文组。

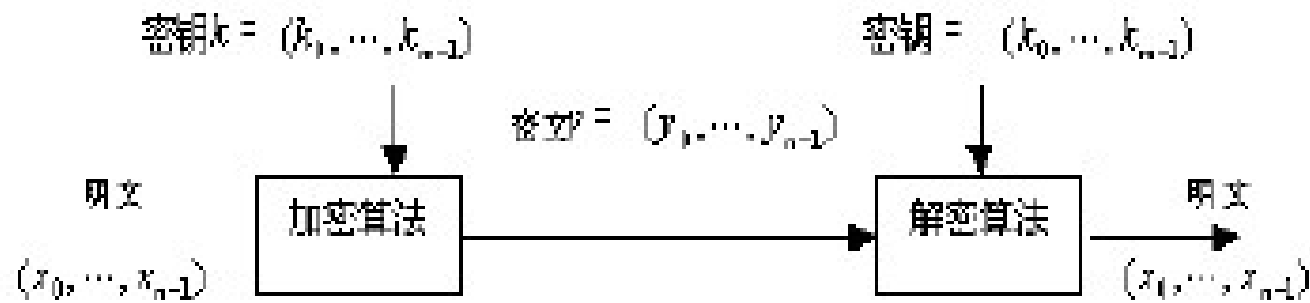
典型分组大小：**64位**或**128位**。

分组密码的应用范围比流密码要广泛。

绝大部分基于网络的对称密码应用使用的是分组密码。

分组密码的一般设计原理：

分组密码是将明文消息编码表示后的数字（简称明文数字）序列，划分成长度为 n 的组（可看成长度为 n 的矢量），每组分别在密钥的控制下变换成等长的输出数字（简称密文数字）序列。



分组密码模型

分组密码概述

- 分组密码是许多系统安全的一个重要组成部分。可用于构造
 - 拟随机数生成器
 - 流密码
 - 消息认证码 (MAC) 和杂凑函数
 - 消息认证技术、数据完整性机构、实体认证协议以及单钥数字签身体制的核心组成部分。

应用中对于分组码的要求

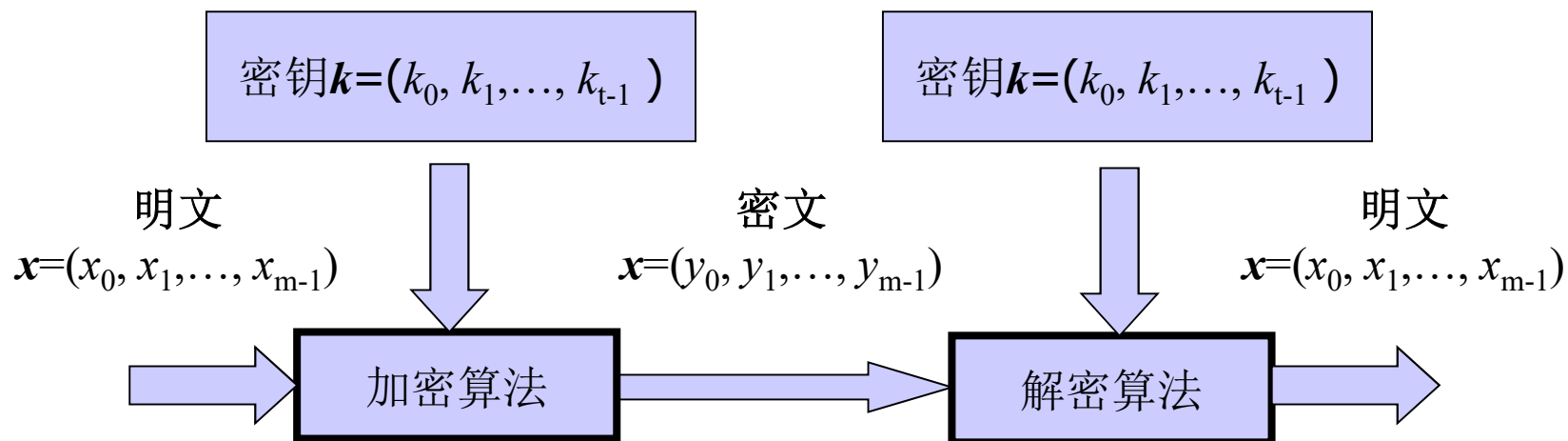
- 安全性
- 运行速度
- 存储量(程序的长度、数据分组长度、高速缓存大小)
- 实现平台(硬、软件、芯片)
- 运行模式

分组密码概述

明文序列 $x_1, x_2, \dots, x_i, \dots$

加密函数 $E: V_n \times K \rightarrow V_n$

这种密码实质上是字长为 m 的数字序列的代换密码。



分组密码概述

- 通常取 $n=m$ 。
- 若 $n>m$ ，则为有数据扩展的分组密码。
- 若 $n<m$ ，则为有数据压缩的分组密码。



分组密码设计问题

分组密码的设计问题在于找到一种算法，能在密钥控制下从一个足够大且足够好的置换子集中，简单而迅速地选出一个置换，用来对当前输入的明文的数字组进行加密变换。

分组密码算法应满足的要求

- 分组长度 n 要足够大：
防止明文穷举攻击奏效。
- 密钥量要足够大：
尽可能消除弱密钥并使所有密钥同等地好，以防止
密钥穷举攻击奏效。
- 由密钥确定置换的算法要足够复杂：
充分实现明文与密钥的扩散和混淆，没有简单的关系可循，要能抗击各种已知的攻击。

分组密码算法应满足的要求

- 加密和解密运算简单：
易于软件和硬件高速实现。
- 数据扩展：
一般无数据扩展，在采用同态置换和随机化加密技术时可引入数据扩展。
- 差错传播尽可能地小。

分组密码的运行模式

即使有了安全的分组密码算法，也需要采用适当的工作模式来隐蔽明文的统计特性、数据的格式等，以提高整体的安全性，降低删除、重放、插入和伪造成功的机会。

- 电子码本 (ECB)
- 密码反馈链接 (CBC)
- 密码反馈 (CFB)
- 输出反馈 (OFB) 。

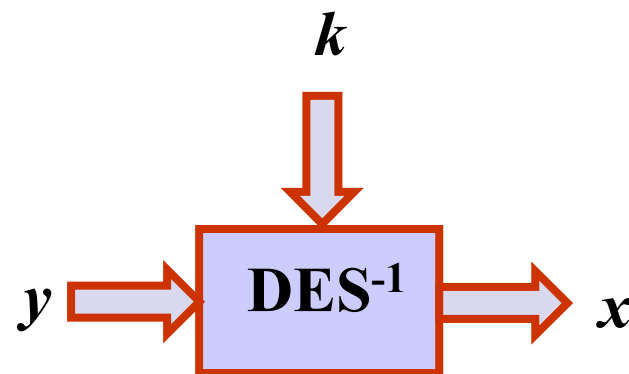
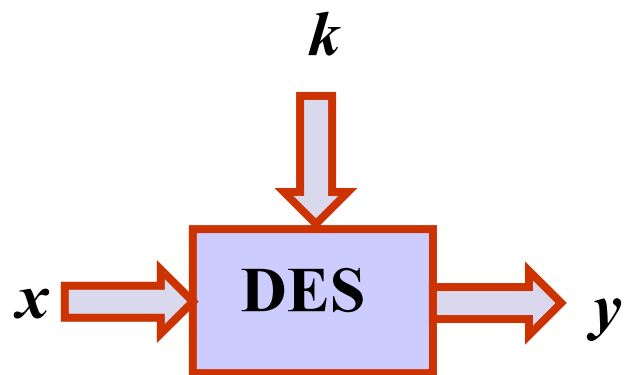


电码本ECB模式

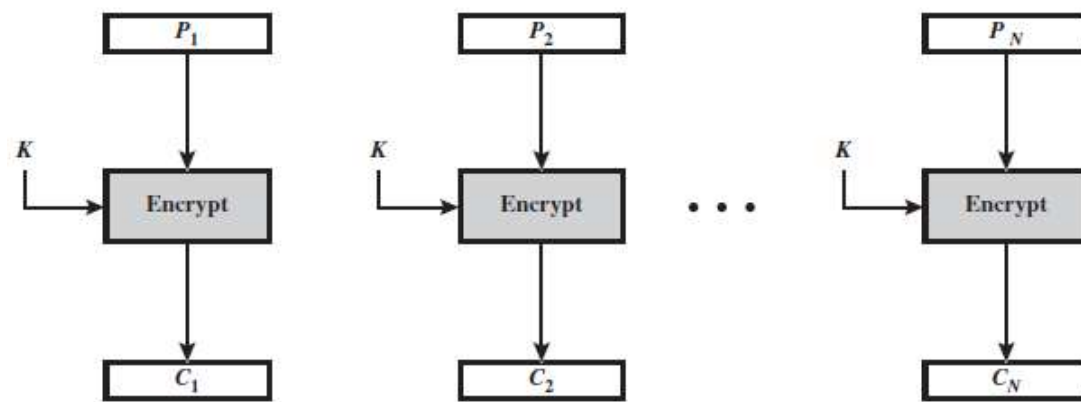
- 直接利用加密算法分别对分组数据组加密。
- 在给定的密钥下同一明文组总产生同样的密文组。这会暴露明文数据的格式和统计特征。

明文数据都有固定的格式，需要以协议的形式定义，重要的数据常常在同一位置上出现，使密码分析者可以对其进行统计分析、重传和代换攻击。

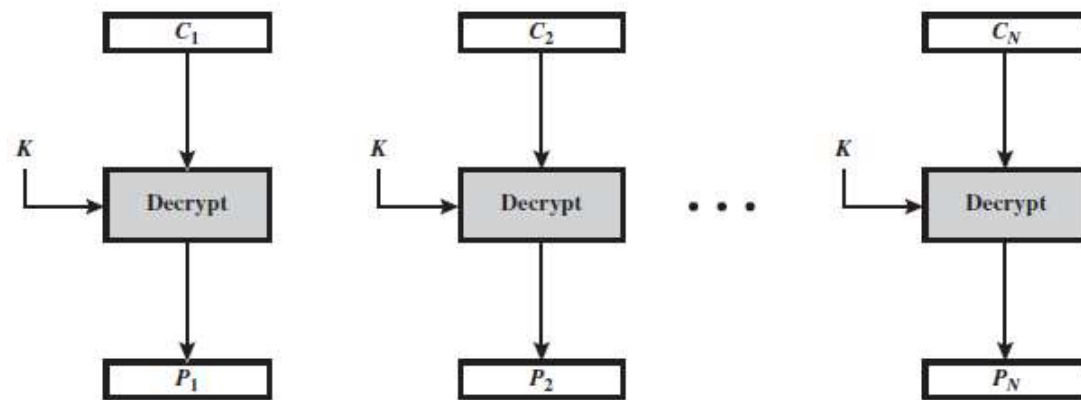
电码本ECB模式



电码本ECB模式

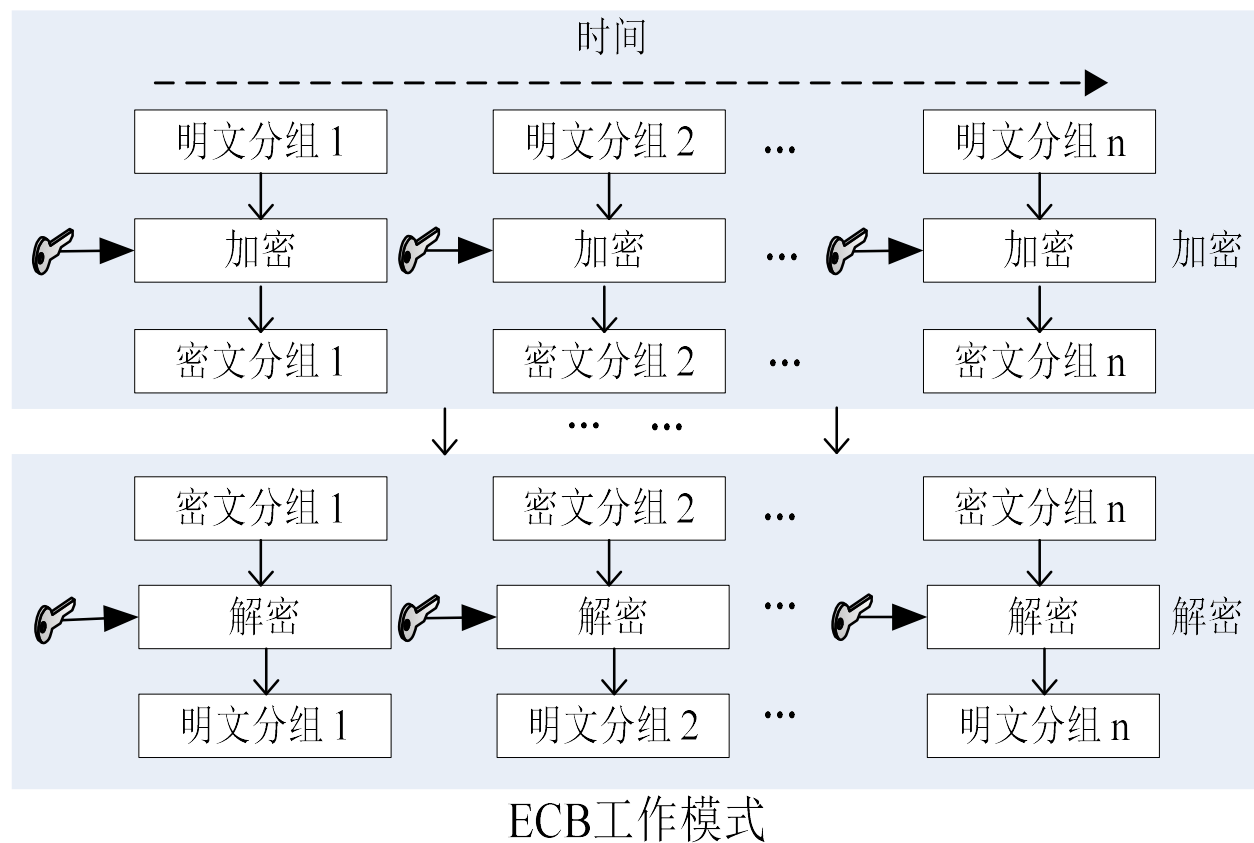


(a) Encryption



(b) Decryption

● 电子编码本模式（ECB）



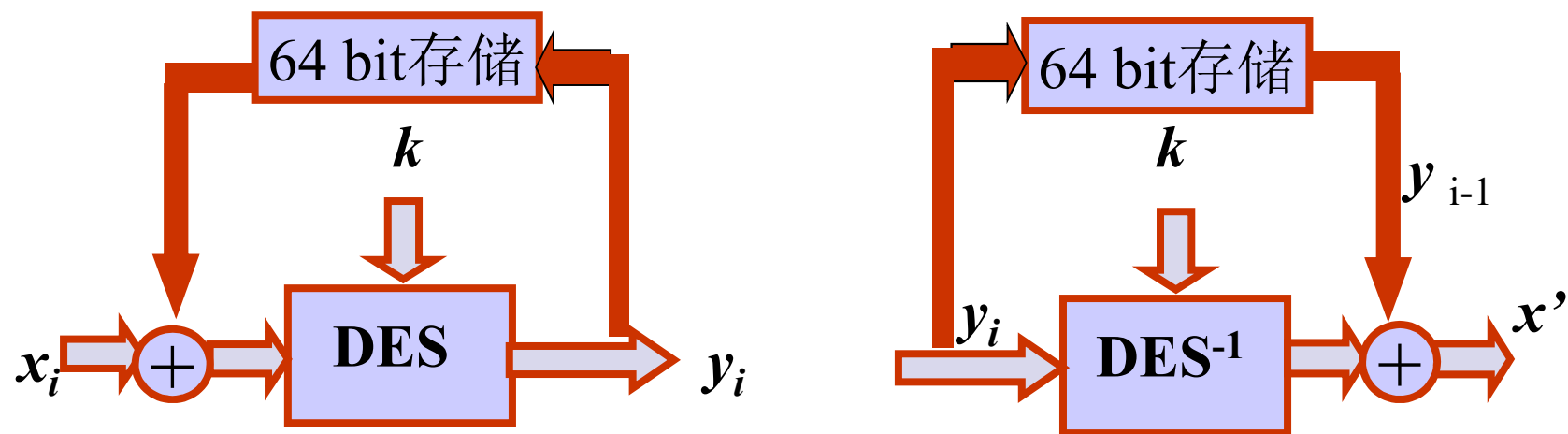
密码分组链接CBC模式

- 每个明文组 x_i 加密之前，先与反馈至输入端的前一组密文 y_{i-1} 按位模2求和后，再送至加密算法加密
- 各密文组 y_i 不仅与当前明文组 x_i 有关，而且通过反馈作用还与以前的明文组 x_1, x_2, \dots, x_{i-1} ，有关

密码分组链接CBC模式

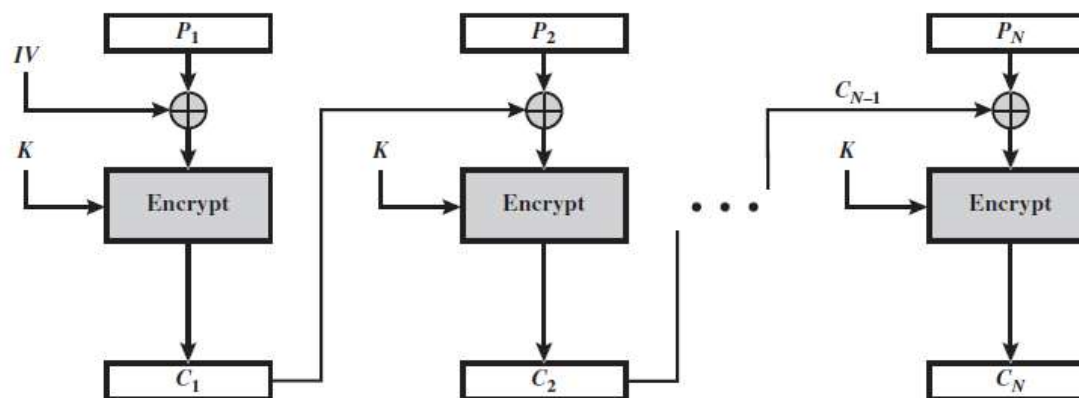
- 初始矢量 IV (Initial Vector): 第一组明文 x_i 加密时尚无反馈密文, 为此需要在寄存器中预先置入一个。收发双方必须选用同一 IV 。
- 实际上, IV 的完整性要比其保密性更为重要。在CBC模式下, 最好是每发一个消息, 都改变 IV , 比如将其值加一。

密码分组链接CBC模式

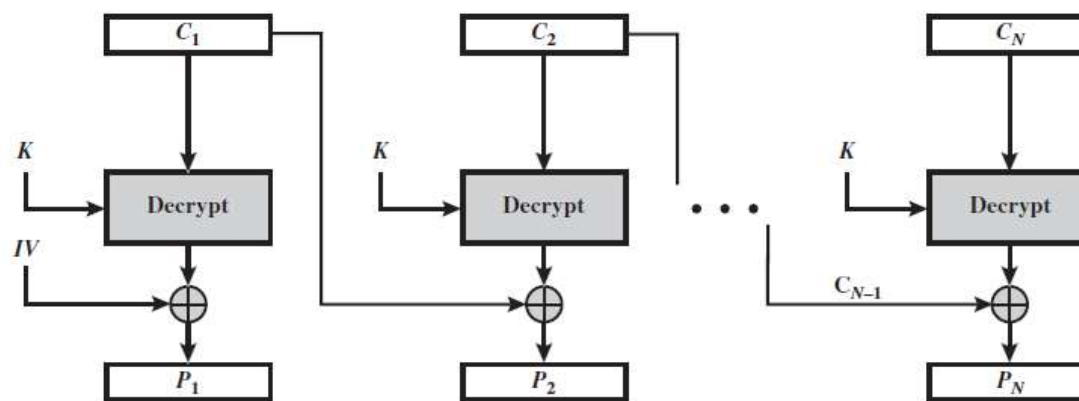


CBC模式

密码分组链接CBC模式



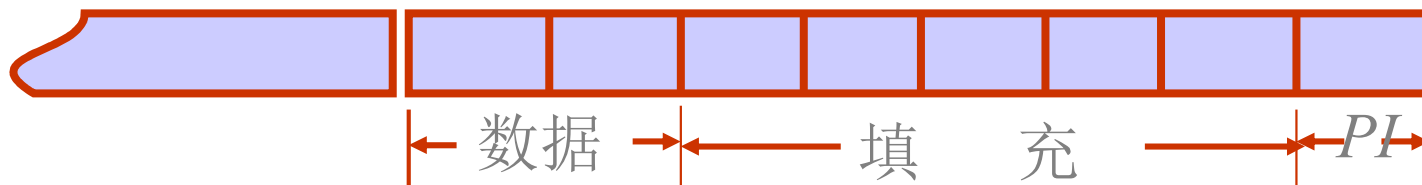
(a) Encryption



(b) Decryption

填充(Padding)

给定加密消息的长度是随机的，按**64 bit**分组时，最后一组消息长度可能不足**64 bit**。可以填充一些数字，通常用最后1字节作为填充指示符（*PI*）。它所表示的十进制数字就是填充占有的字节数。数据尾部、填充字符和填充指示符一起作为一组进行加密。





CBC的错误传播

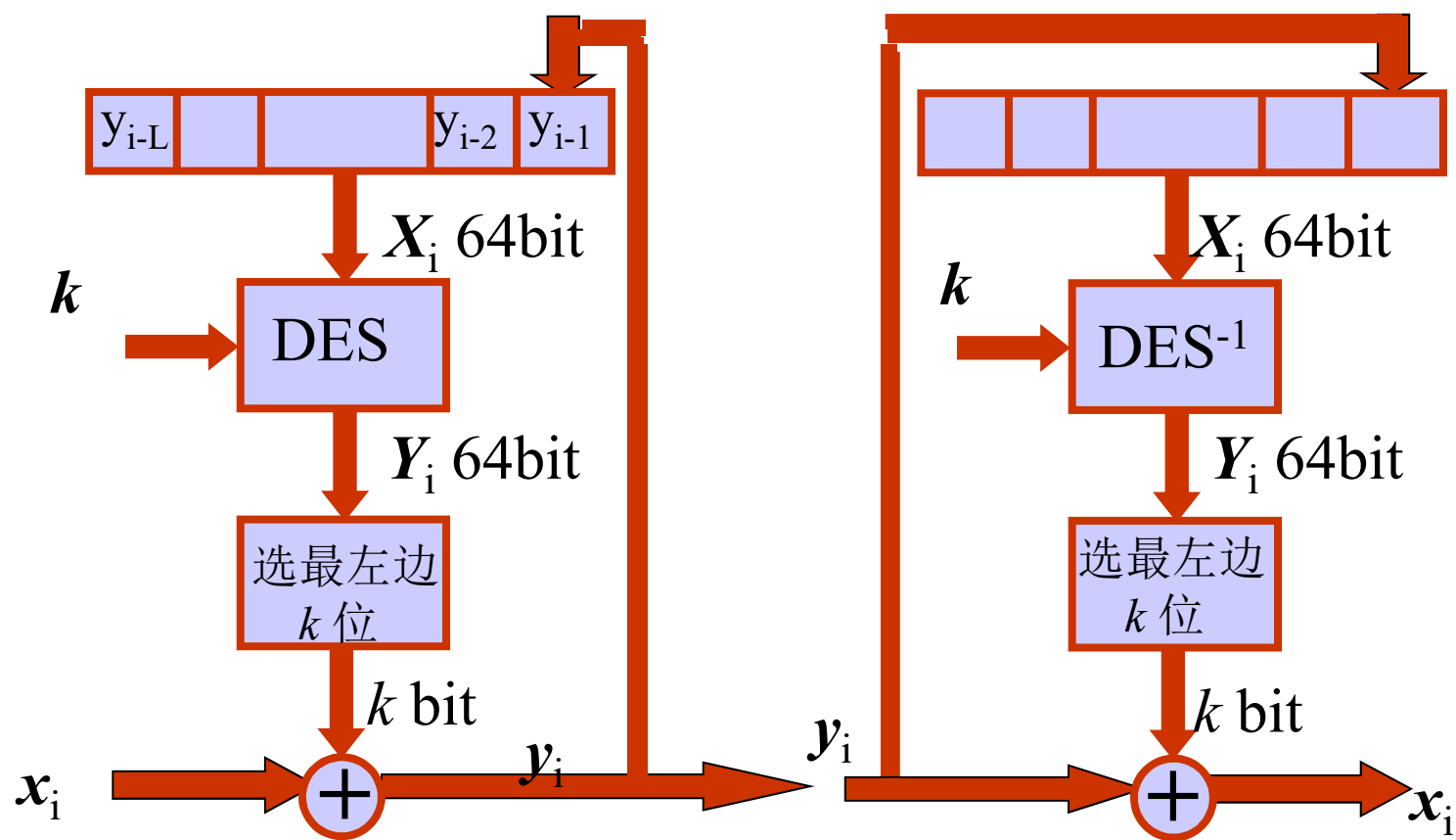
1. 明文有一组中有错，会使以后的密文组都受影响，但经解密后的恢复结果，除原有误的一组外，其后各组明文都正确地恢复。
2. 若在传送过程中，某组密文组 y_i 出错时，则该组恢复的明文 x'_i 和下一组恢复数据 x'_{i+1} 出错。再后面的组将不会受 y_i 中错误比特的影响。



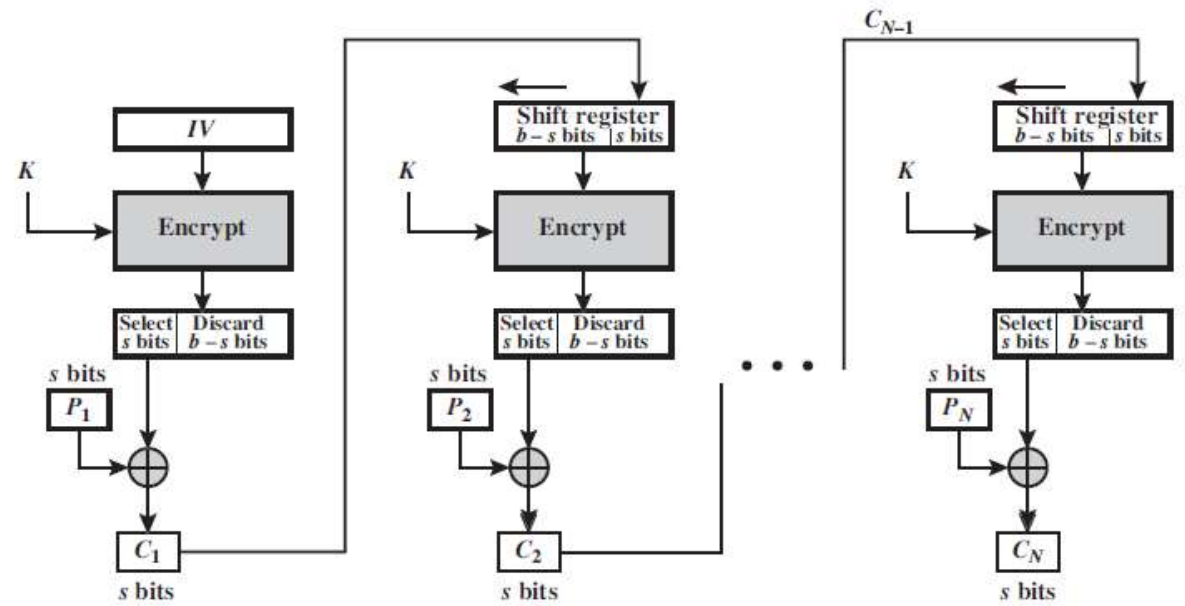
k -比特密码反馈CFB模式

- 若待加密消息必须按字符(如电传电报)或按比特处理时，可采用**CFB**模式。
- **CFB**实际上是将加密算法**DES**作为一个密钥流产生器，当 $k=1$ 时就退化为前面讨论的流密码了。
- **CFB**与**CBC**的区别是反馈的密文长度为 k ，且不是直接与明文相加，而是反馈至密钥产生器。

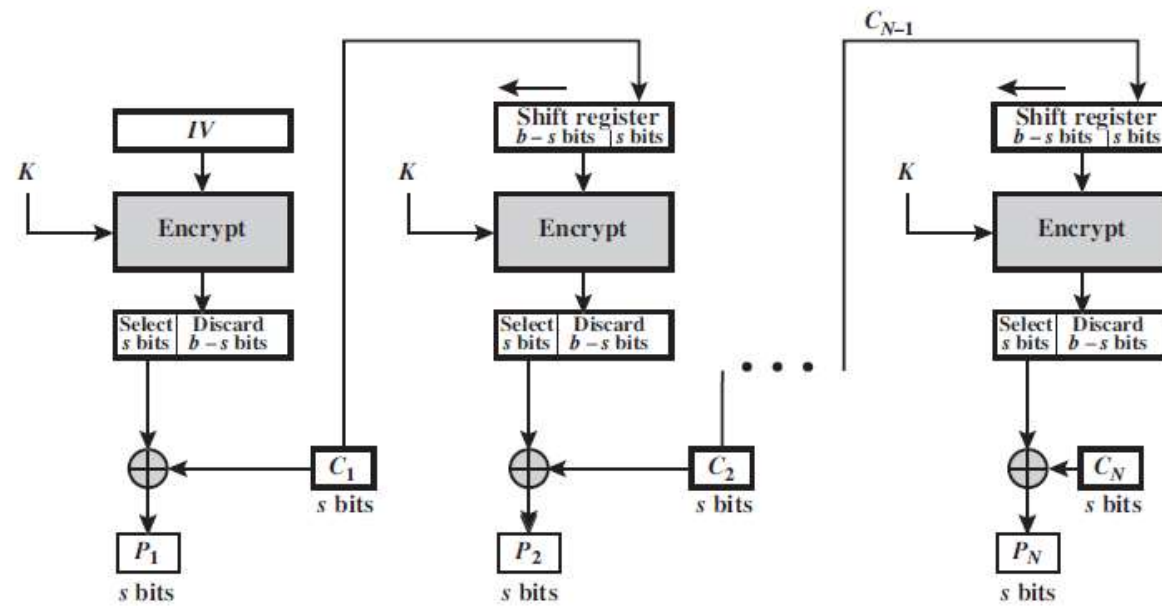
k -比特密码反馈CFB模式



CFB模式

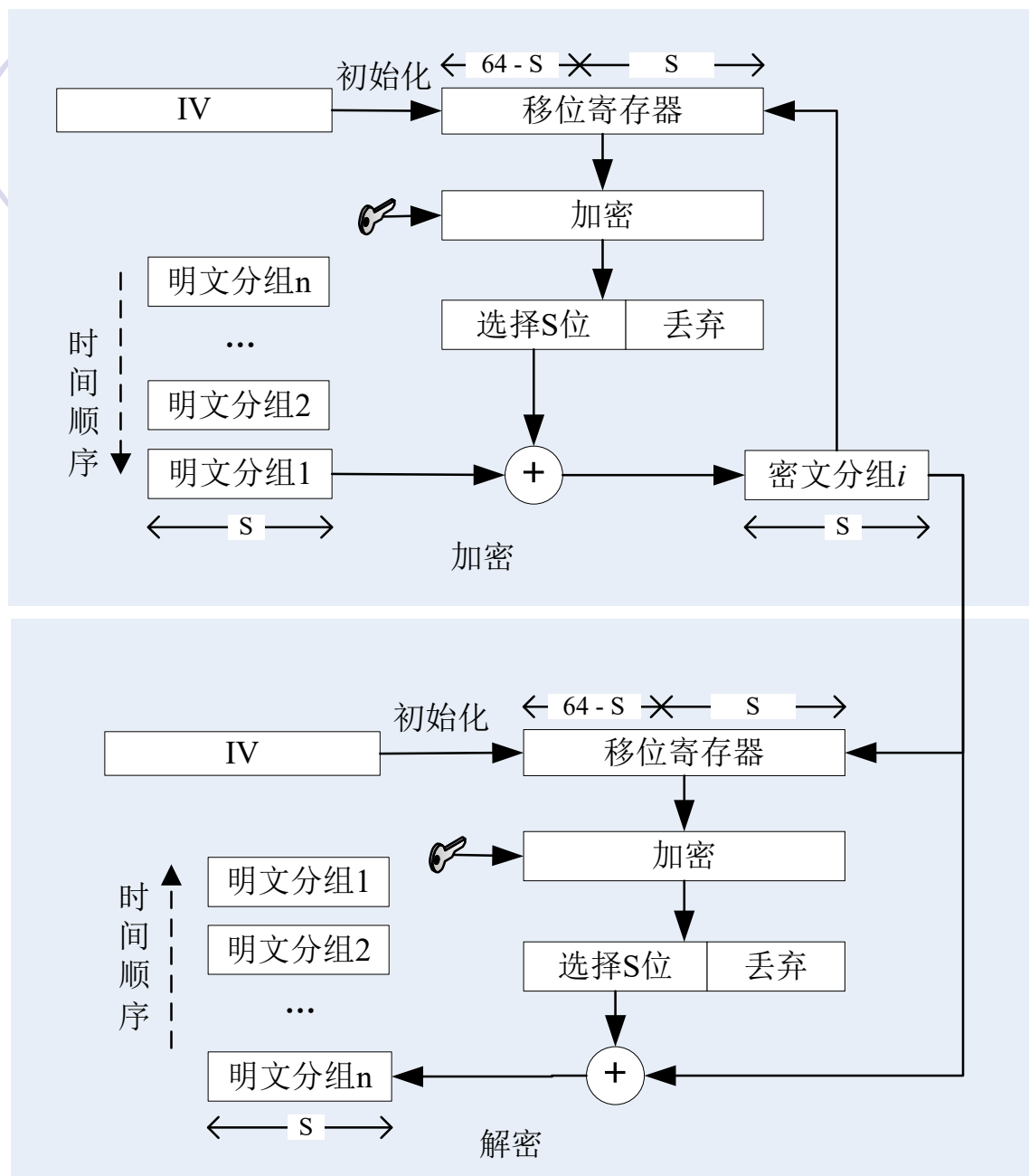


(a) Encryption



(b) Decryption

密码反馈模式 (CFB)



CFB 工作模式



k -比特密码反馈CFB模式

- **CFB**的优点

- 它特别适于用户数据格式的需要。
- 能隐蔽明文数据图样，也能检测出对手对于密文的篡改。

- **CFB**的缺点

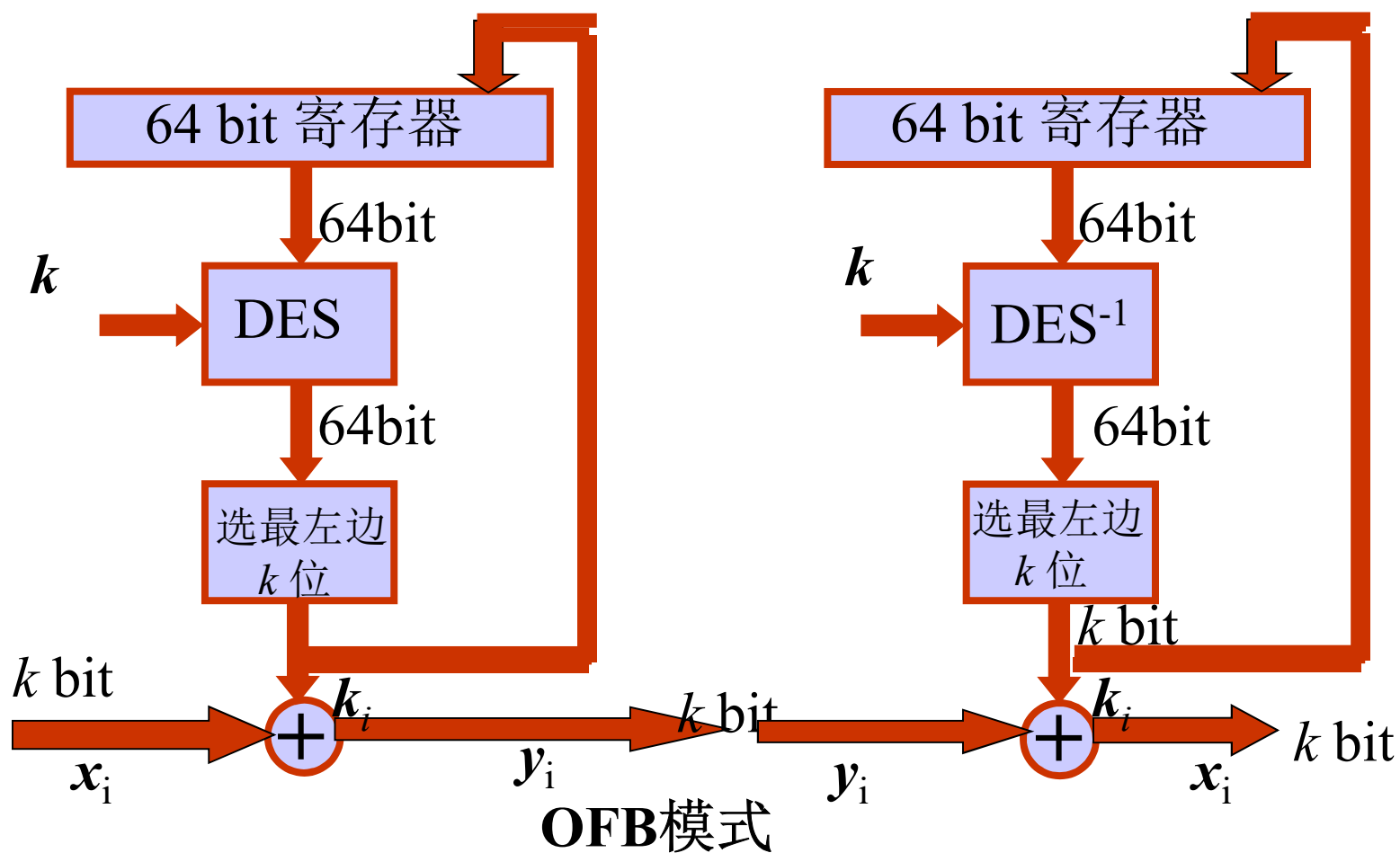
- 对信道错误较敏感，且会造成错误传播。
- **CFB**也需要一个初始矢量，并要和密钥同时进行更换。

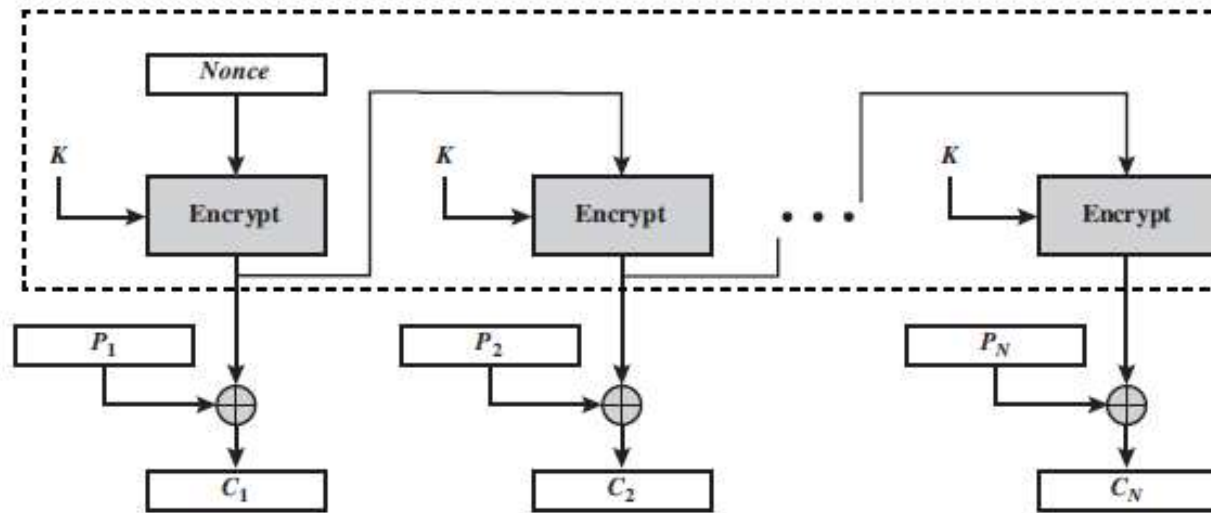


输出反馈OFB模式

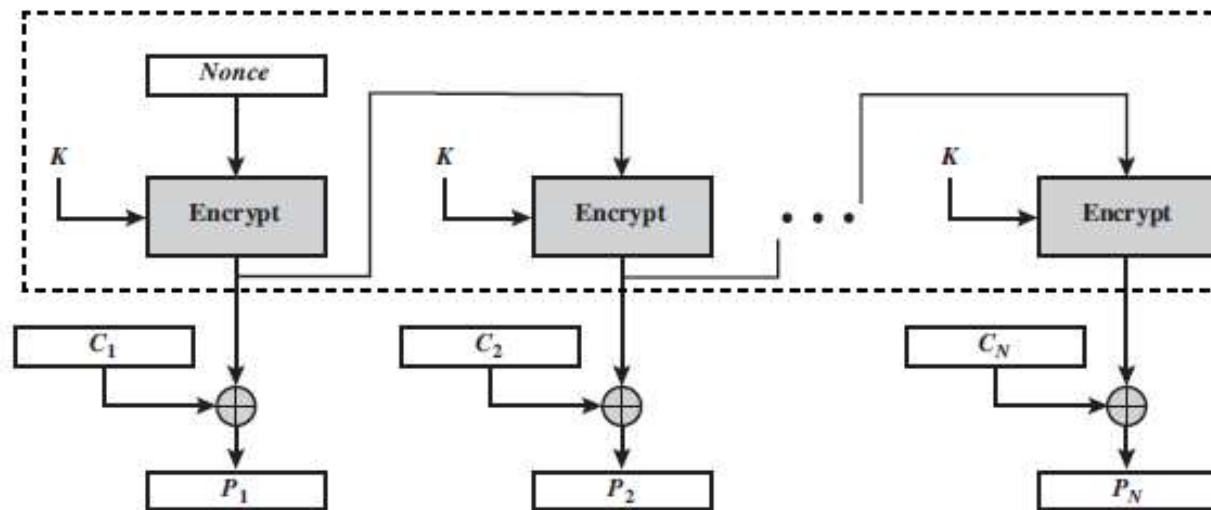
- 将分组密码算法作为一个密钥流产生器，其输出的 k -bit密钥直接反馈至分组密码的输入端，同时这 k -bit密钥和输入的 k -bit明文段进行对应位模2相加。
- 克服了CBC和CFB的错误传播所带来的问题。
- 对于密文被篡改难以进行检测
- 不具有自同步能力，要求系统要保持严格的同步

输出反馈OFB模式





(a) Encryption



(b) Decryption

输出反馈模式 (OFB)

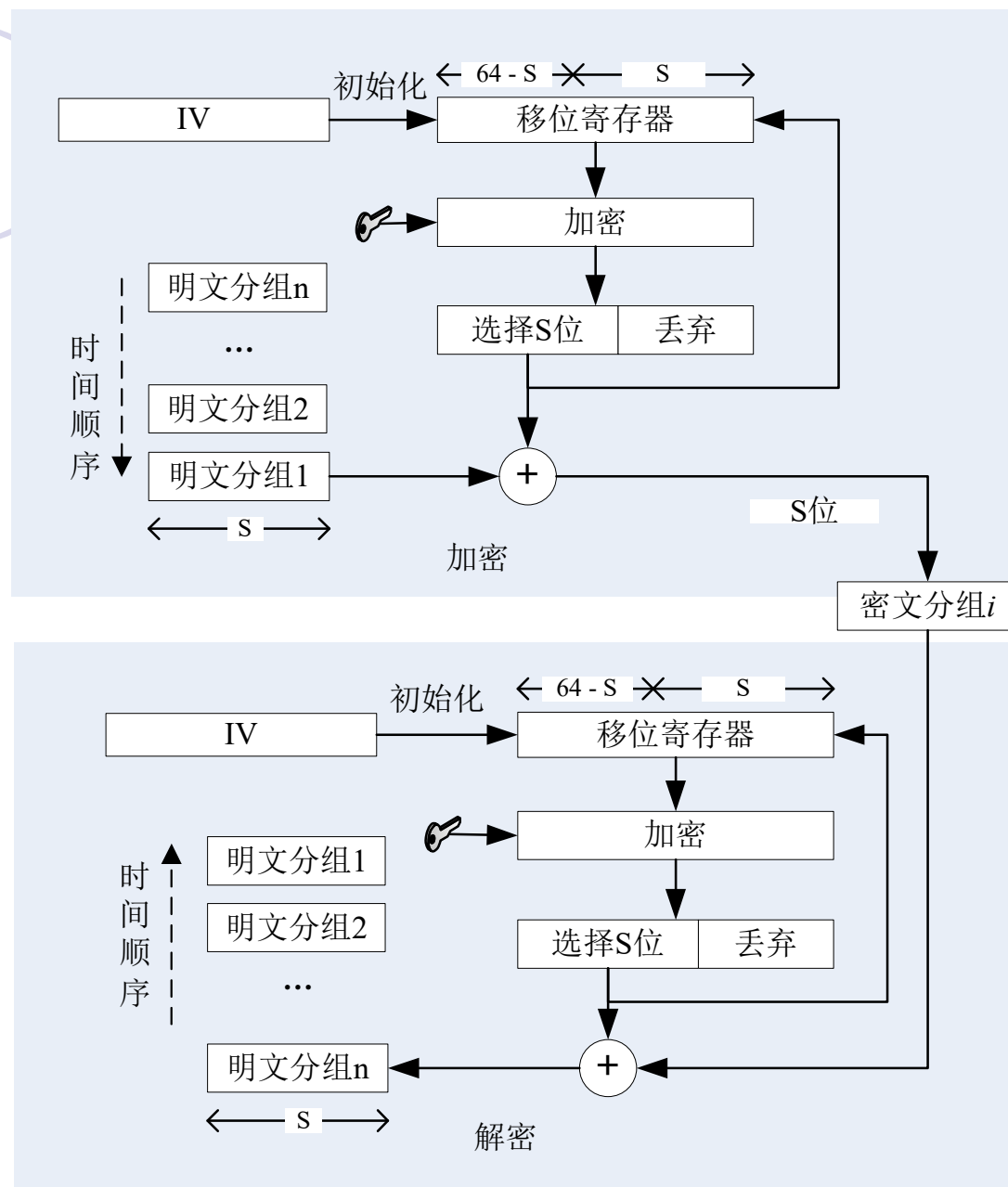


图 2.14 OFB 工作模式



比较和选用

- **ECB**模式，简单、高速，但最弱、易受重发攻击，一般不推荐。
- **CBC**适用于文件加密，但较**ECB**慢。安全性加强。当有少量错误时，也不会造成同步错误。
- **OFB**和**CFB**较**CBC**慢许多。每次迭代只有少数bit完成加密。若可以容忍少量错误扩展，则可换来恢复同步能力，此时用**CFB**。在字符为单元的流密码中多选**CFB**模式。
- **OFB**用于高速同步系统，不容忍差错传播。

美国数据加密标准—DES (Data Encryption Standard)

- 目的

通信与计算机相结合是人类步入信息社会的一个阶梯，它始于六十年代末，完成于90年代初。计算机通信网的形成与发展，要求信息作业标准化，安全保密亦不例外。只有标准化，才能真正实现网的安全，才能推广使用加密手段，以便于训练、生产和降低成本。

美国制定数据加密标准简况

- 美国NBS在1973年5月15公布了征求建议。1974年8月27日NBS再次出公告征求建议，对建议方案提出如下要求：
- 算法必须完全确定而无含糊之处；
- 算法必须有足够高的保护水准，即可以检测到威胁，恢复密钥所必须的运算时间或运算次数足够大；
- 保护方法必须只依赖于密钥的保密；
- 对任何用户或产品供应者必须是不加区分的。

美国制定数据加密标准简况

- IBM公司在1971年完成的LUCIFER密码 (64 bit分组, 代换-置换, 128 bit密钥)的基础上, 改进成为建议的DES体制
- 1975年3月17日NBS公布了这个算法, 并说明要以它作为联邦信息处理标准, 征求各方意见。
- 1977年1月15日建议被批准为联邦标准[FIPS PUB 46], 并设计推出DES芯片。
- 1981年美国ANSI 将其作为标准, 称之为DEA[ANSI X3.92]
- 1983年国际标准化组织(ISO)采用它作为标准, 称作DEA-1

美国制定数据加密标准简况

- NSA宣布每隔5年重新审议DES是否继续作为联邦标准，1988年（FIPS46-1）、1993年（FIPS46-2），1998年不再重新批准DES为联邦标准。
- 虽然DES已有替代的数据加密标准算法，但它仍是迄今为止得到最广泛应用的一种算法，也是一种最有代表性的分组加密体制。
- 1993年4月，Clinton政府公布了一项建议的加密技术标准，称作密钥托管加密技术标准EES(Escrowed Encryption Standard)。算法属美国政府SECRET密级。

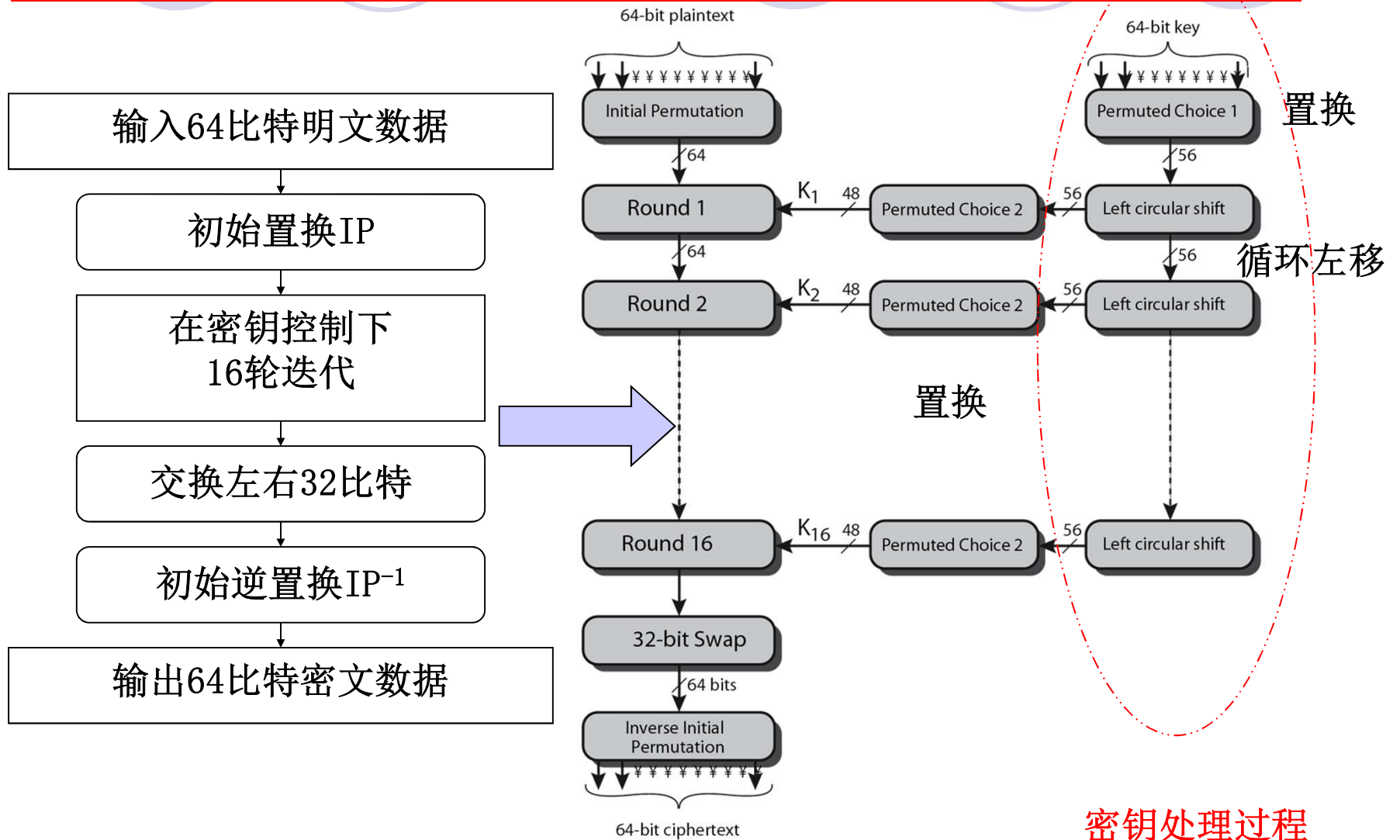
美国制定数据加密标准简况

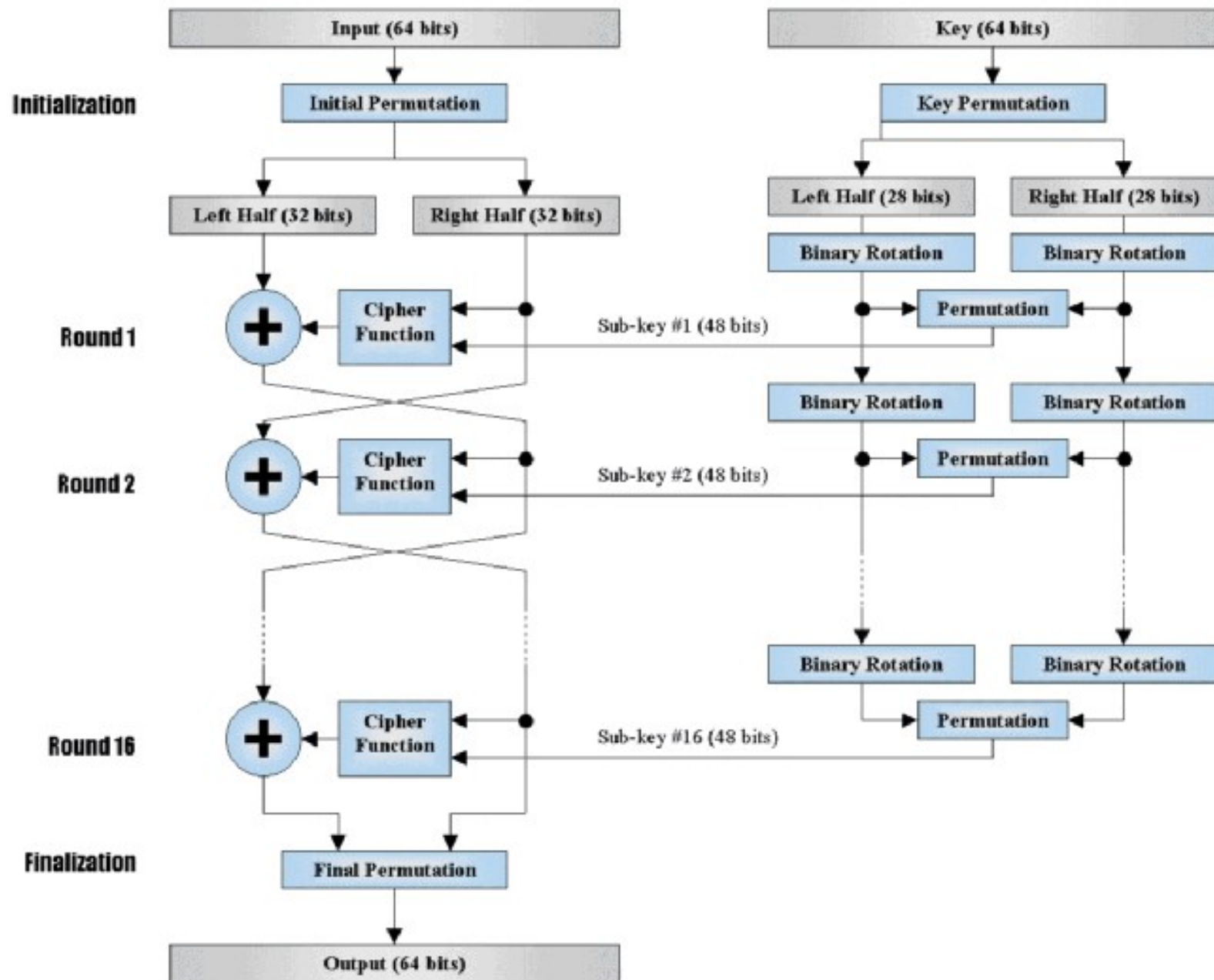
- **DES**发展史确定了发展公用标准算法模式，而**EES**的制定路线与**DES**的背道而驰。人们怀疑有陷门和政府部门肆意侵犯公民权利。此举遭到广为反对。
- **1995年5月AT&T Bell Lab的M. Blaze博士在PC机上用45分钟时间使SKIPJACK的 LEAF协议失败，伪造ID码获得成功。1995年7月美国政府宣布放弃用EES来加密数据，只将它用于语音通信。**
- **1997年1月美国NIST着手进行AES（Advanced Encryption Standard）的研究，成立了标准工作室。2001年Rijndael被批准为AES标准。**



- 分组长度为**64 bits (8 bytes)**
- 密文分组长度也是**64 bits**。
- 密钥长度为**64 bits**，有**8 bits**奇偶校验，有效密钥长度为**56 bits**。
- 算法主要包括：初始置换 **IP** 、**16**轮迭代的乘积变换、逆初始置换 **IP^{-1}** 以及**16**个子密钥产生器。

DES Encryption Overview

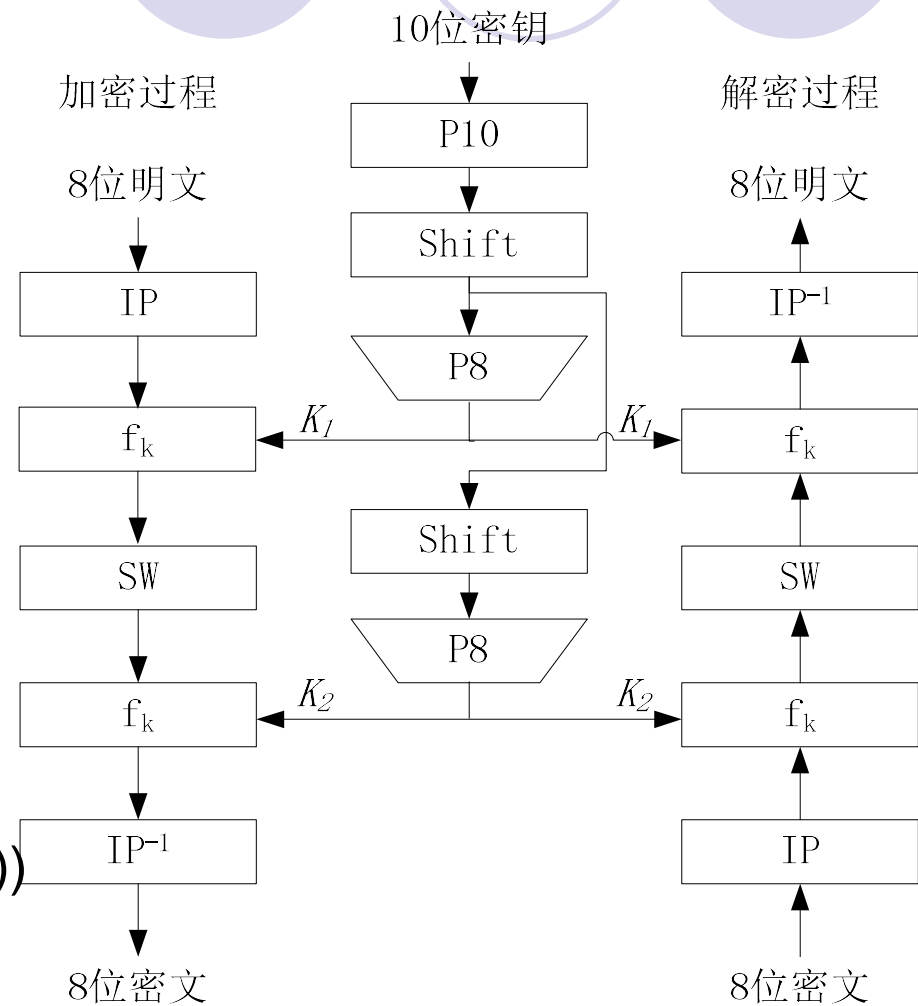




S-DES

● 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： $IP^{-1}(f_{k_2}(SW(f_{k_1}(IP(明文)))))$
- 解密： $IP^{-1}(f_{k_1}(SW(f_{k_2}(IP(密文)))))$

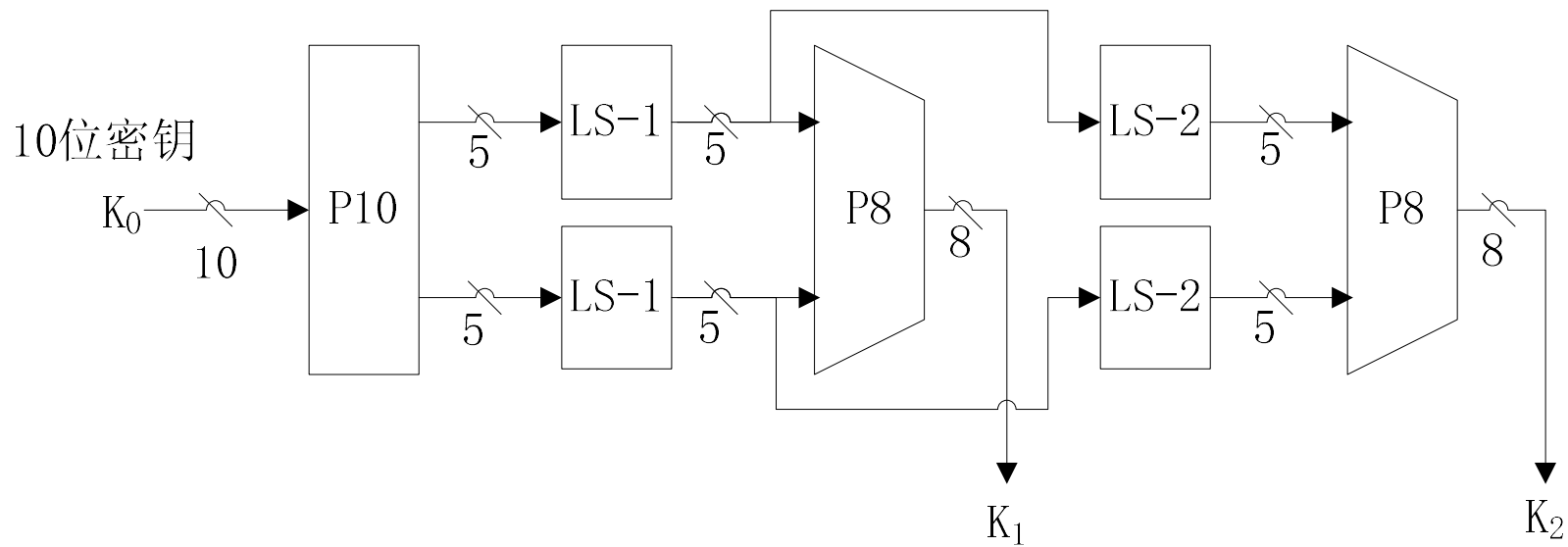


S-DES的体制

- P10 = (3, 5, 2, 7, 4, 10, 1, 9, 8, 6)

- ## ○循环左移函数LS

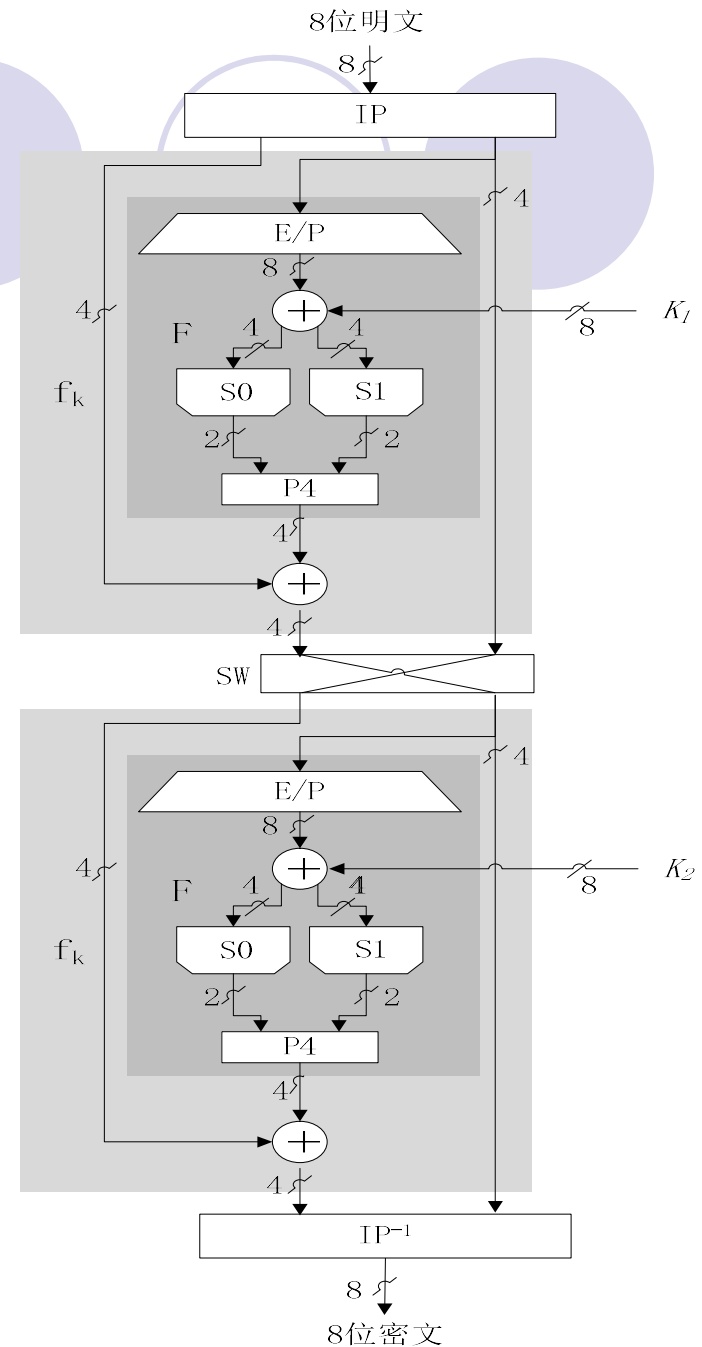
- P8=(6,3,7,4,8,5,10,9)



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



The diagram shows four circles arranged horizontally. The first circle is filled with light purple. The second circle is empty with a light purple outline. The third circle is filled with light purple. The fourth circle is empty with a light purple outline. Below the first circle is the S0 matrix, and below the third circle is the S1 matrix.

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

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

- S盒函数按下述规则运算：

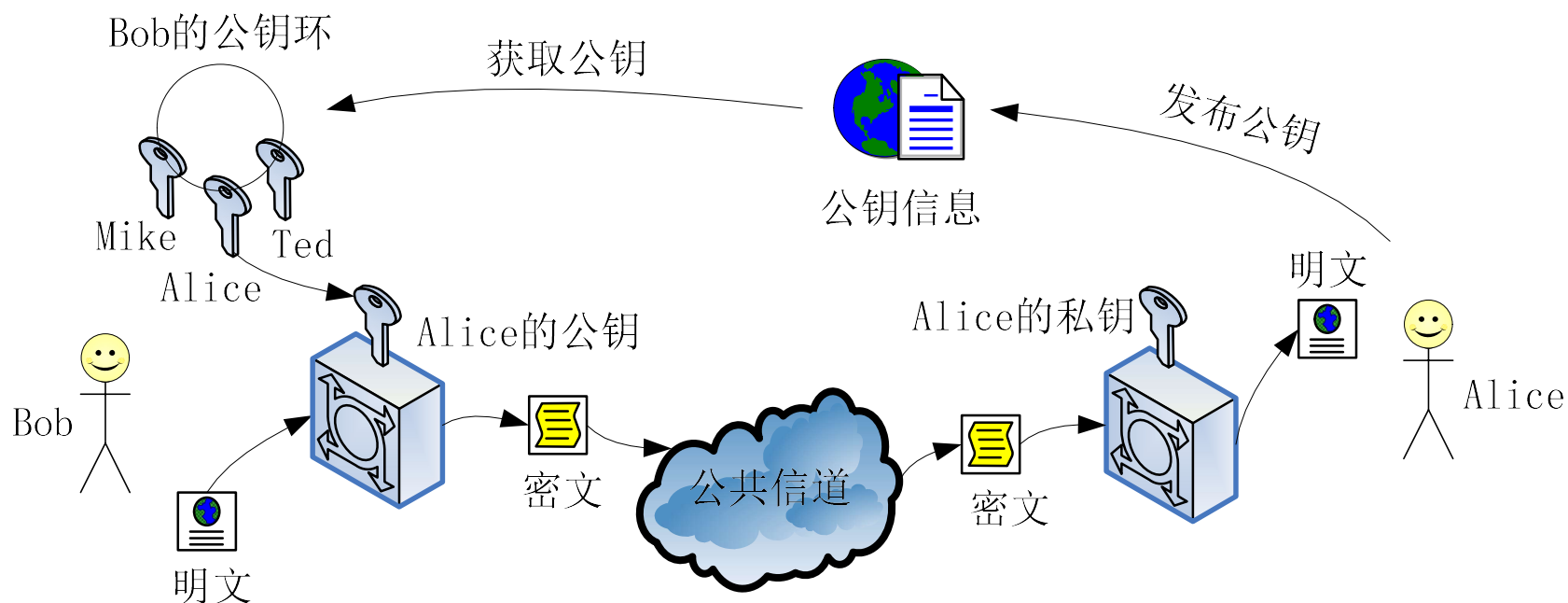
- 输入的第1位和第4位二进制数合并为一个两位二进制数，作为S盒的行号索引*i*；
- 将第2位和第3位同样合并为一个两位二进制数，作为S盒的列号索引*j*，
- 确定S盒矩阵中的一个系数 (i, j) 。
- 此系数以两位二进制数形式作为S盒的输出。
- 例如：
 - $L' = (l_0, l_1, l_2, l_3) = (0, 1, 0, 0)$, $(i, j) = (0, 2)$
 - 在S0中确定系数3，则S0的输出为11B。

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。



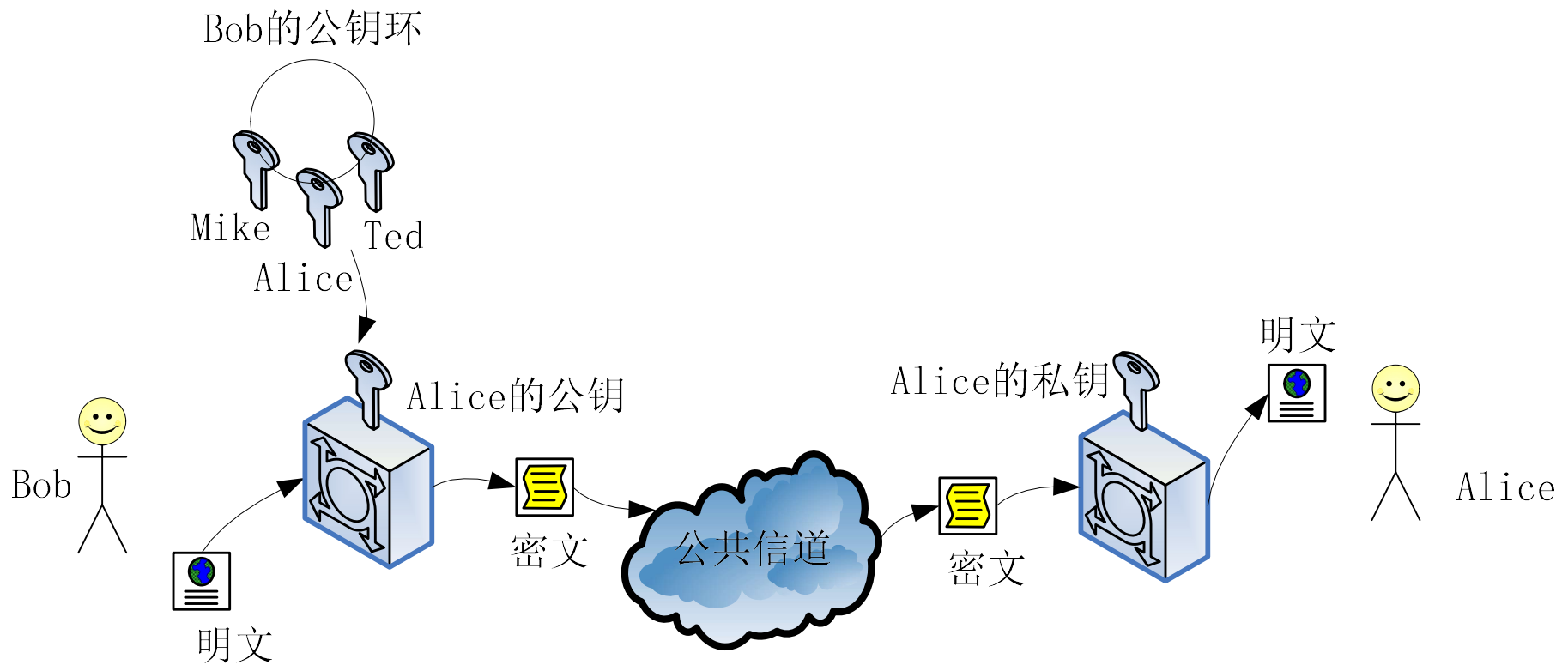
公开密钥密码的模型

2.4.1 公开密钥理论基础

公开密钥密码的核心思想

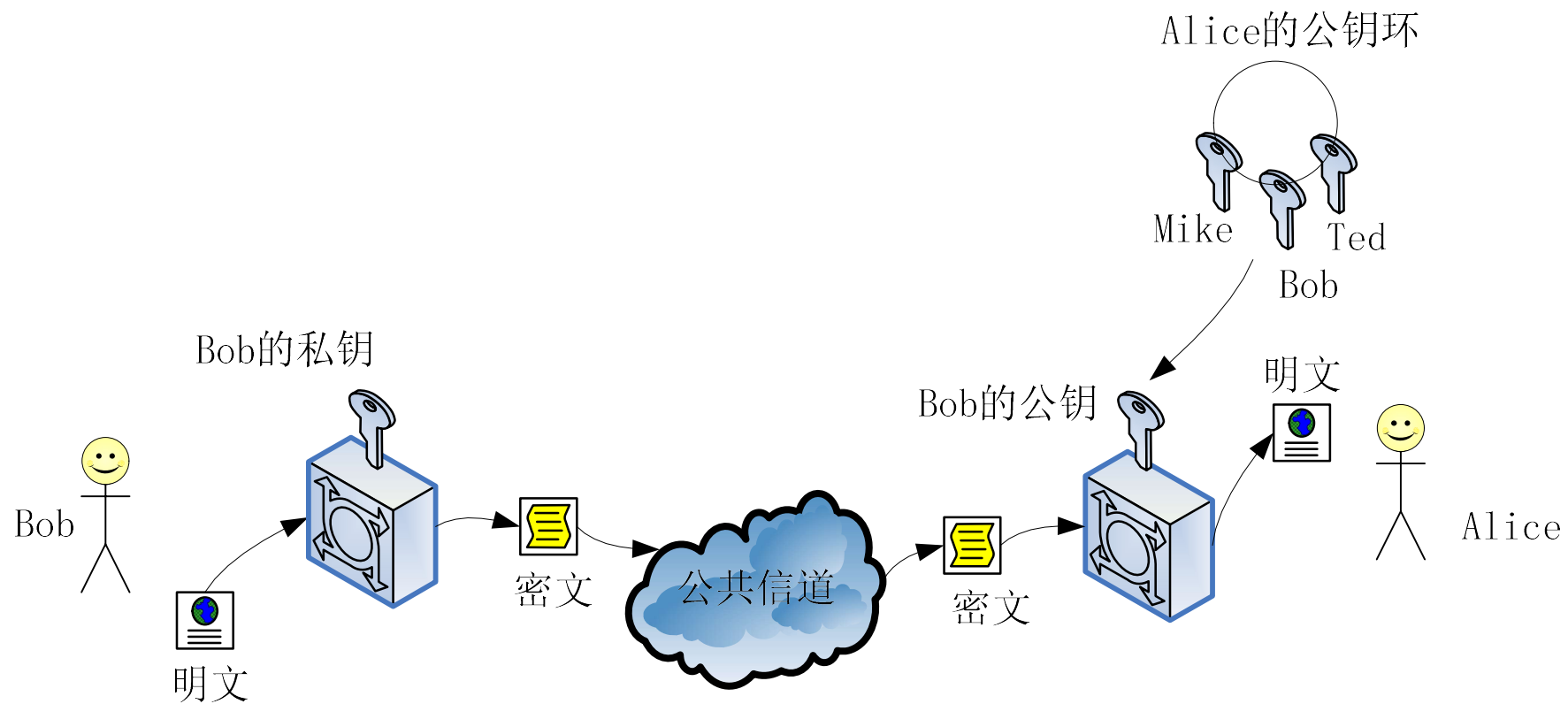
- 公开密钥密码是1976年由Whitfield Diffie和Martin Hellman在其“密码学新方向”一文中提出的。
- 单向陷门函数 $f(x)$ ，必须满足以下三个条件。
 - ① 给定 x ，计算 $y=f(x)$ 是容易的；
 - ② 给定 y ，计算 x 使 $y=f(x)$ 是困难的（所谓计算 $x=f^{-1}(y)$ 困难是指计算上相当复杂已无实际意义）；
 - ③ 存在 δ ，已知 δ 时对给定的任何 y ，若相应的 x 存在，则计算 x 使 $y=f(x)$ 是容易的。

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 $\varphi(n)$ ，称为欧拉函数，其中 $\varphi(1)$ 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 $\varphi(n) = (p-1)(q-1)$ ；
- 欧拉定理的具体表述：正整数 a 与 n 互素，则 $a^{\varphi(n)} \equiv 1 \pmod n$ 。

● 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， $m^{k\varphi(n)+1} = m^{k(p-1)(q-1)+1} \equiv m \pmod n$ 。

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：
 - 是一种分组密码，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 $C=\{x \in Z | 0 < x < n, Z \text{ 为整数集合}\}$ 。
- 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 $Pk = \{ e, n \}$ ，私钥 $Sk = \{ d, n, p, q \}$

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，
- 消息内容是否曾受到偶然或有意的篡改，
- 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

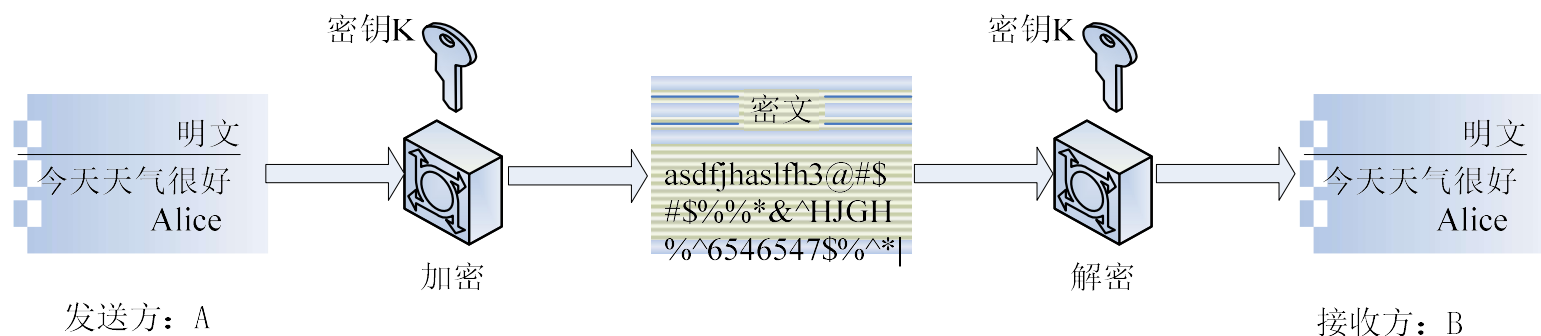
- 消息加密，用消息的完整密文作为消息的认证符；
- 消息认证码**MAC**（**Message Authentication Code**），也称密码校验和，使用密码对消息加密，生成固定长度的认证符；
- 消息编码，是针对信源消息的编码函数，使用编码抵抗针对消息的攻击。

2.5.2 认证函数

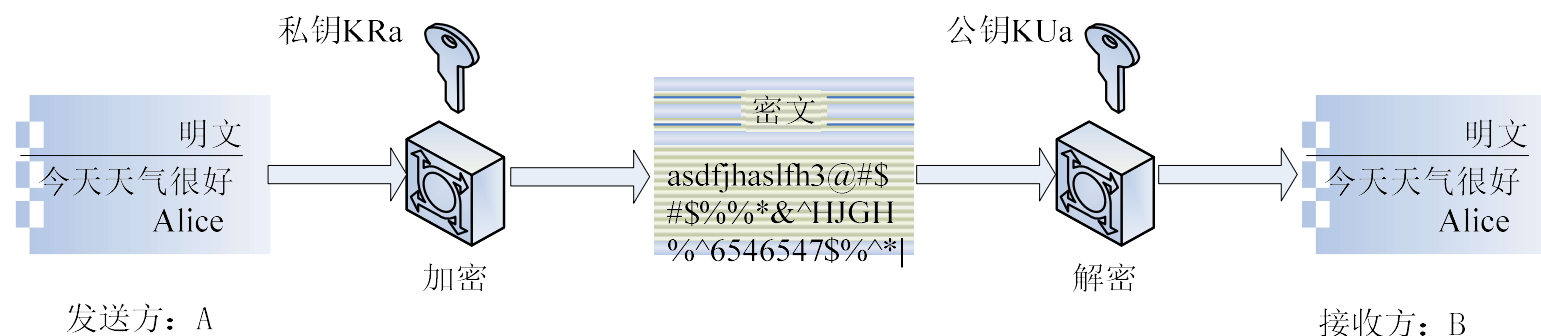
- 认证技术在功能上可以分为两层
 - 下层包含一个产生认证符的函数，认证符是一个用来认证消息的值；
 - 上层是以认证函数为原语，接收方可以通过认证函数来验证消息的真伪。

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

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

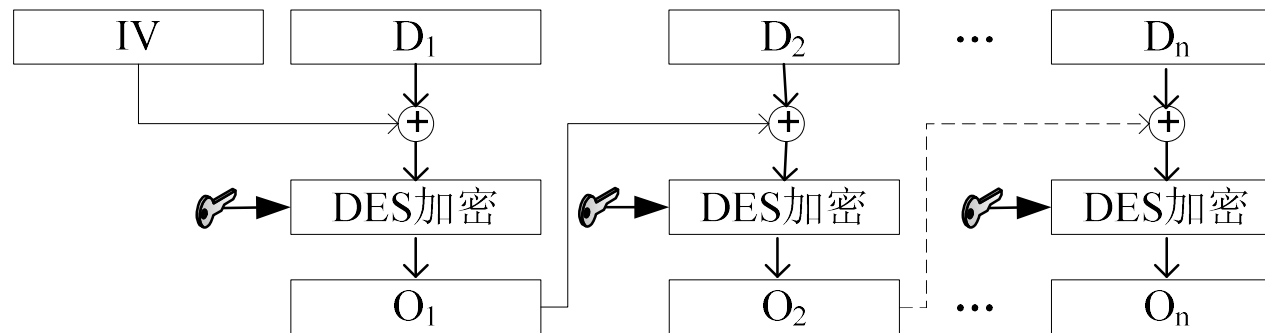



图2.19 基于DES的消息认证码

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%**。

○ 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。

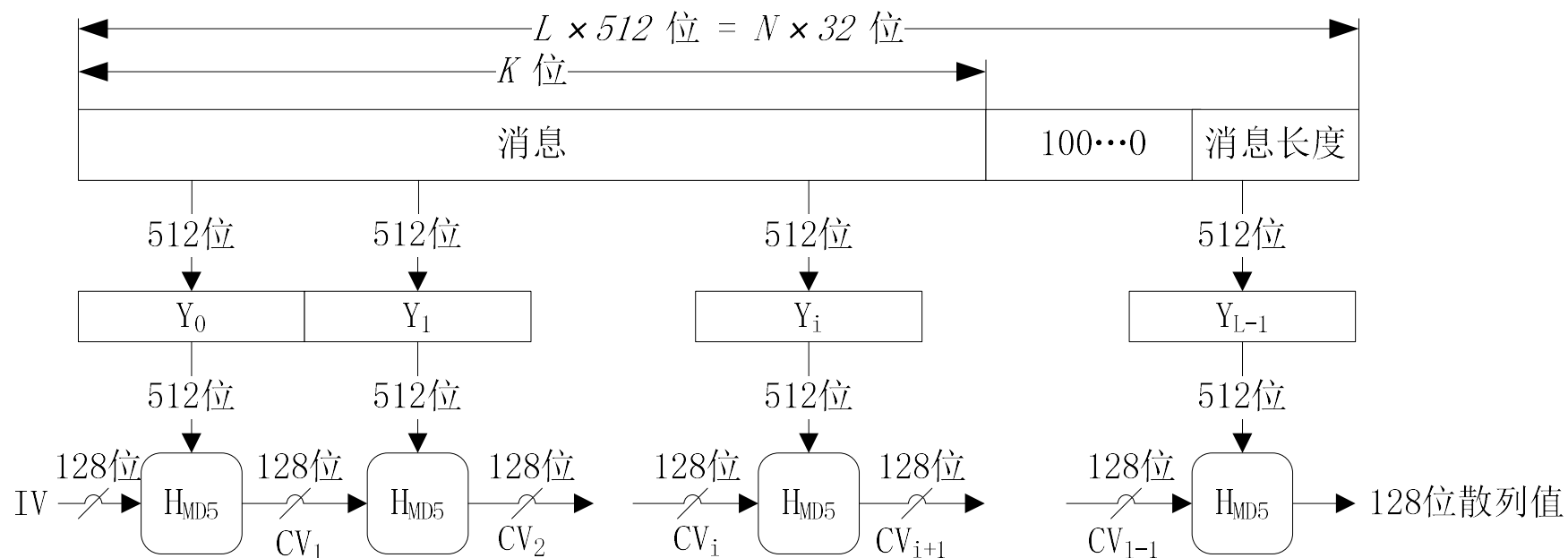


图2.20 MD5算法

MD5

- 四轮运算涉及四个函数：

- $E(X, Y, Z) = (X \wedge Y) \vee ((\neg X) \wedge Z)$

- $F(X, Y, Z) = (X \wedge Z) \vee (Y \wedge (\neg Z))$

- $G(X, Y, Z) = X \oplus Y \oplus Z$

- $H(X, Y, Z) = Y \oplus (X \vee (\neg Z))$

- 第一轮：

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

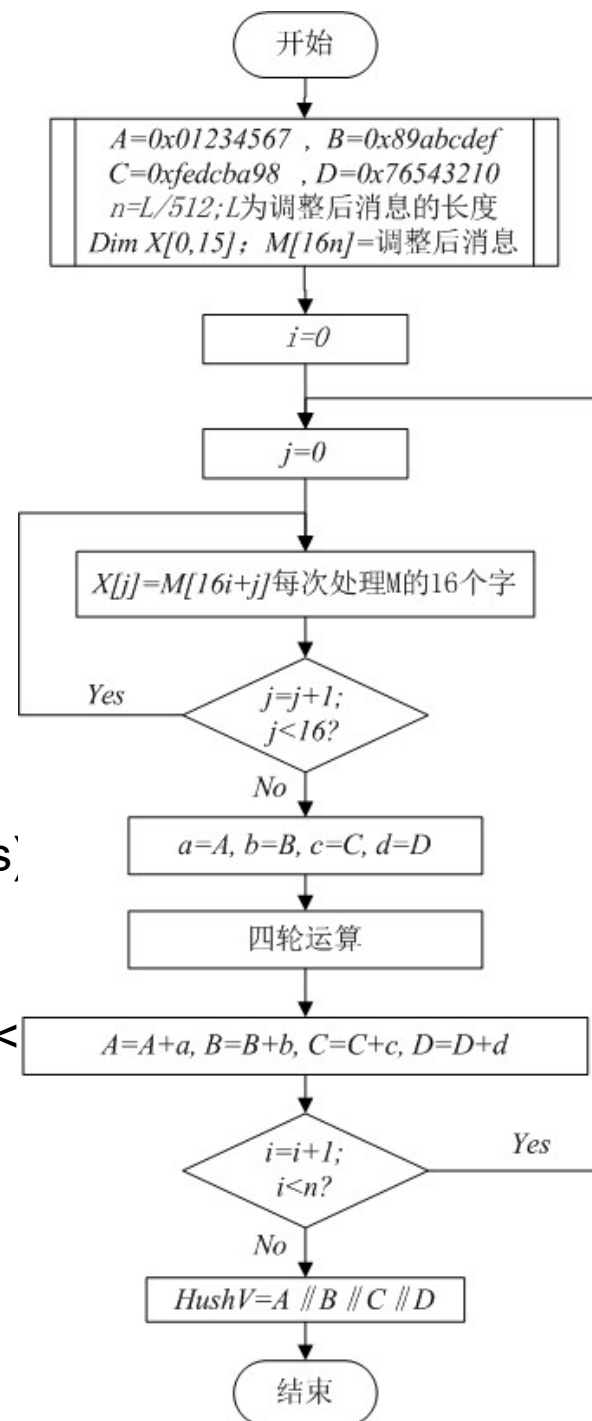
- 第二轮：

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

- 第三轮：

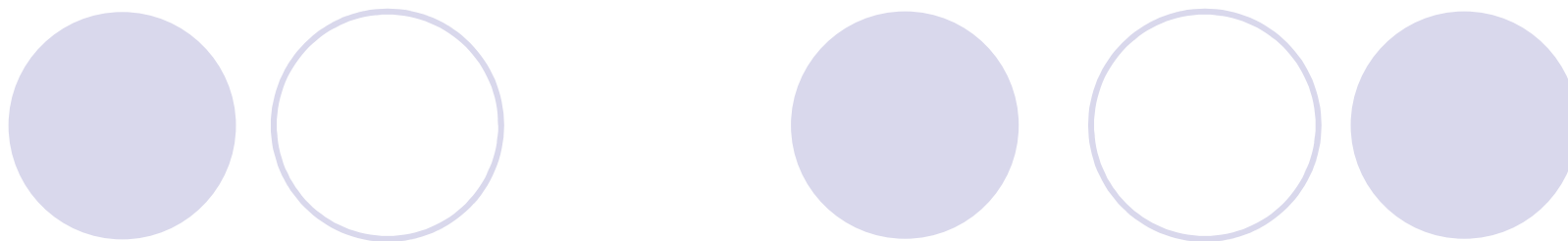
- $GG(a, b, c, d, M_j, s, t_i) : a = b + ((a + (G(b, c, d) + M_j + t_i) << s)$

- 第四轮：



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。

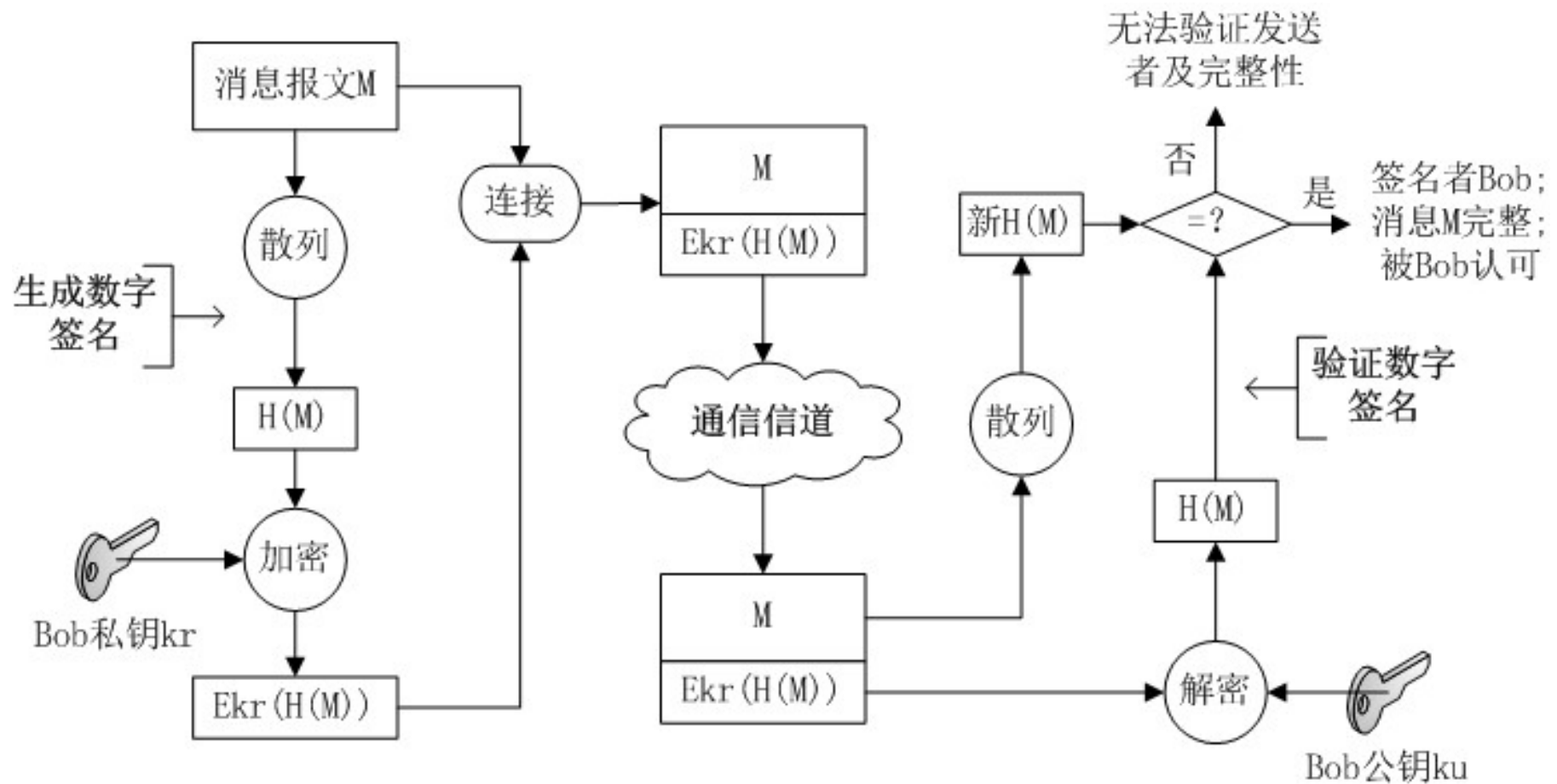


- 从签名形式上分，数字签名有两种
 - 一种是对整个消息的签名，
 - 一种是对压缩消息的签名，
 - 它们都是附加在被签名消息之后或在某一特定位置上的一段数据信息。
- 数字签名主要目的
 - 保证收方能够确认或验证发方的签名，但不能伪造；发方发出签名消息后，不能否认所签发的消息。



- 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。