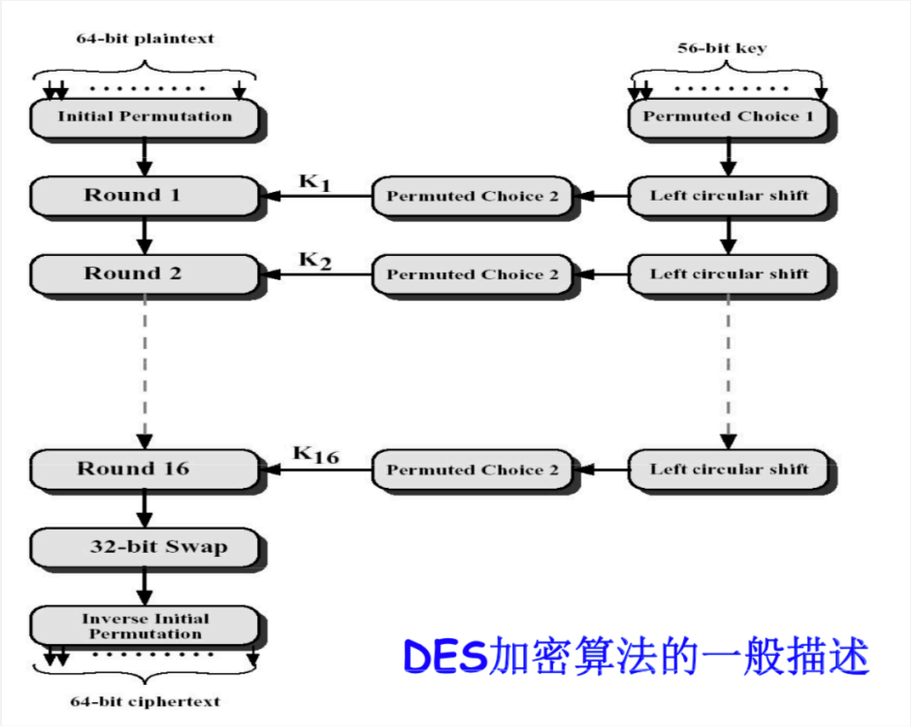
DES

🔪 by kingno

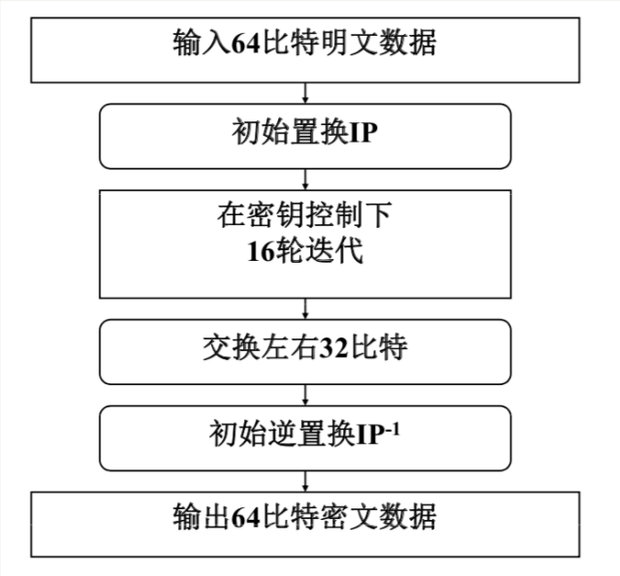
数据加密标准（DES）

DES 概述

- 分组加密算法：明文和密文为64位长度的分组（这意味着，如果是ASCII编码，每个字符占8个比特，那么就是8个字符一组进行加密）
- 对称算法：加密和解密除密钥编排不同外，使用同一算法
- 密钥长度：任意的56位数
- 半弱密钥：用该密钥连续把明文P加密两次会得到明文P，即$E(E(P)) = P$
- 弱密钥：$E_{k_1}(P) = E_{k_2}(P)$，即两个不同的密钥加密明文P，得到的密文相同



DES 加密过程



初始置换 IP

首先，把64比特的明文数据打散成8×8的数据矩阵，其中数字代表对应的比特位

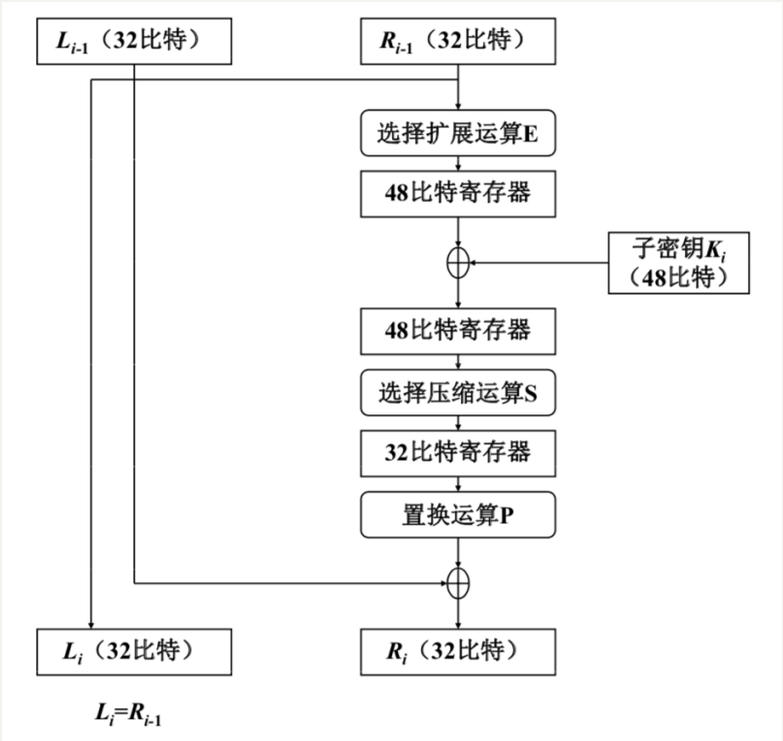
$$\mathrm{M} = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 \\ 17 & 18 & 19 & 20 & 21 & 22 & 23 & 24 \\ 25 & 26 & 27 & 28 & 29 & 30 & 31 & 32 \\ 33 & 34 & 35 & 36 & 37 & 38 & 39 & 40 \\ 41 & 42 & 43 & 44 & 45 & 46 & 47 & 48 \\ 49 & 50 & 51 & 52 & 53 & 54 & 55 & 56 \\ 57 & 58 & 59 & 60 & 61 & 62 & 63 & 64 \end{bmatrix}$$

有一个初始置换矩阵。矩阵中的数据指的是位置，例如58指将明文的第58位放在第1位。M经过IP置换后变为M'

58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

16轮迭代

16轮迭代中，每轮迭代是相同的，所以只需要关注其中的一轮迭代



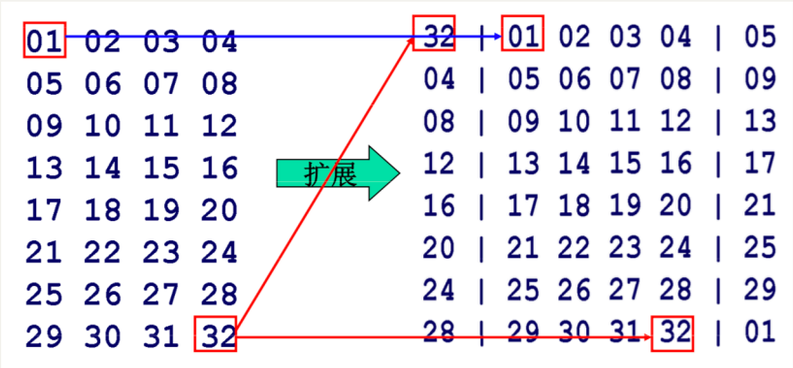
- $L_i \leftarrow R_{i-1}$
- $R_i \leftarrow L_{i-1} \oplus f(R_{i-1})$
- 选择扩展运算E：把32比特的R_{i-1}扩展成48位
- 选择压缩运算S：有8个盒子，每个盒子输入6位，输出4位。所以，总的输入48位，输出32位

选择扩展运算 E

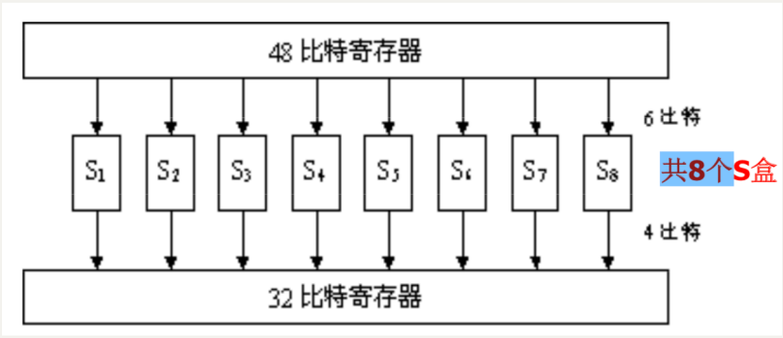
作用：把32比特的R_{i-1}扩展成48位

方法：

- 对于第一列，扩展它的前一个比特。比如，05的前一个比特是04
- 对于最后一列，扩展它的后一个比特。比如，32的后一个比特是01



选择压缩运算 S



一共有8个盒子，每个盒子输入6位，输出4位

对于每一个盒子，它里面有一个 4×16 的矩阵，每行是数字0-15的全排列

对于输入的6个数字$b_1b_2b_3b_4b_5b_6$，b_1b_5表示的二进制数（0-3）用于选择行，$b_2b_3b_4b_5$所表示的二进制（0-15）数用于选择列

比如，下面就是一个盒子中的矩阵

对于输入110011，$b_1b_6 = 11_2 = 3$，$b_2b_3b_4b_5 = 1001_2 = 9$，所以选择的数字是$14$，输出为$14$的二进制表示$1100_2$

S-盒的构造：

- DES中其他算法都是线性的，而S-盒运算是非线性的
- S-盒不易于分析，提供了更好的安全性
- S-盒是算法的关键所在

S-盒的构造准则（扩展内容，看PPT）

置换运算 P

里面是一个4×8的矩阵，做和 ***初始置换IP*** 相似的变换

- P置换的目的是提供雪崩效应，即明文或密文的一点小的变动都会引起密文较大的变化

子密钥 K_i 的生成

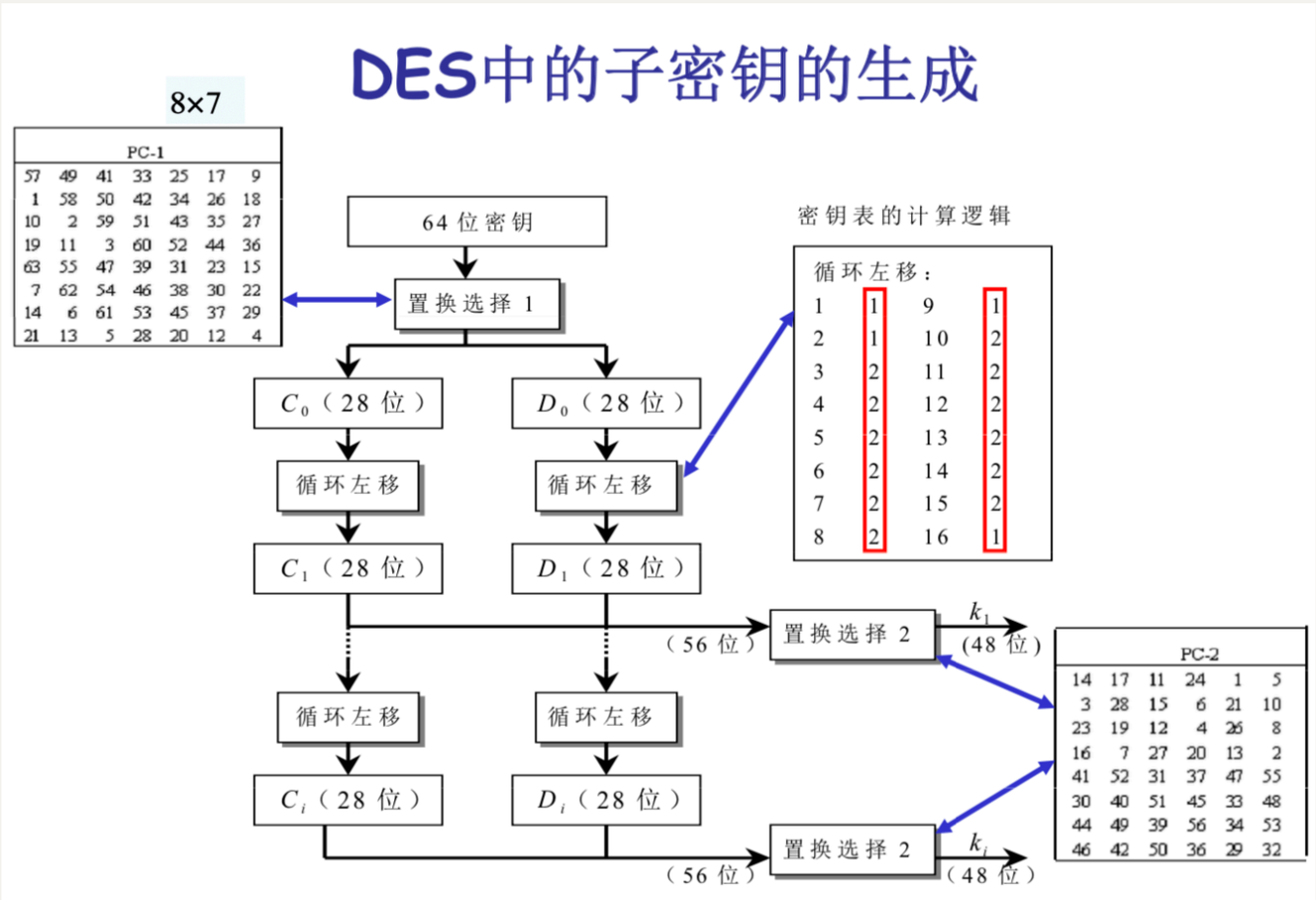
密钥是56位的，而子密钥K_i是48位的

虽然密钥是56位的，但是存储的时候是按8×8的矩阵存储，其中，每行的最后一个比特是 *奇校验位*

- 利用7×8 的 $PC-1$矩阵对8×8的密钥矩阵进行变换。仔细观察$PC-1$矩阵发现，里面的数字不能被8整除，这意味着，是密钥中的数字在置换，奇校验位不置换。变换后得到K'
- 取K'的前28位为C_0，后28位为D_0（别管奇校验位！）
- 每轮移动的移动位数在 ***密钥表的计算逻辑*** 中。比如，第一轮左移1位，第3轮左移

2位

- 第一轮左移得到C_1, D_1，将它们两个合并，经过$PC-2$置换得到K_1
- 每轮左移都得到一个K_i
- 还要经过一个置换选择，将56位变为48位
- 奇偶校验位：用于在网络传输的过程中，检测是否有比特差错



交换左右32比特

因为中间密文是64位的，所以这步操作是把左32比特和右32比特交换

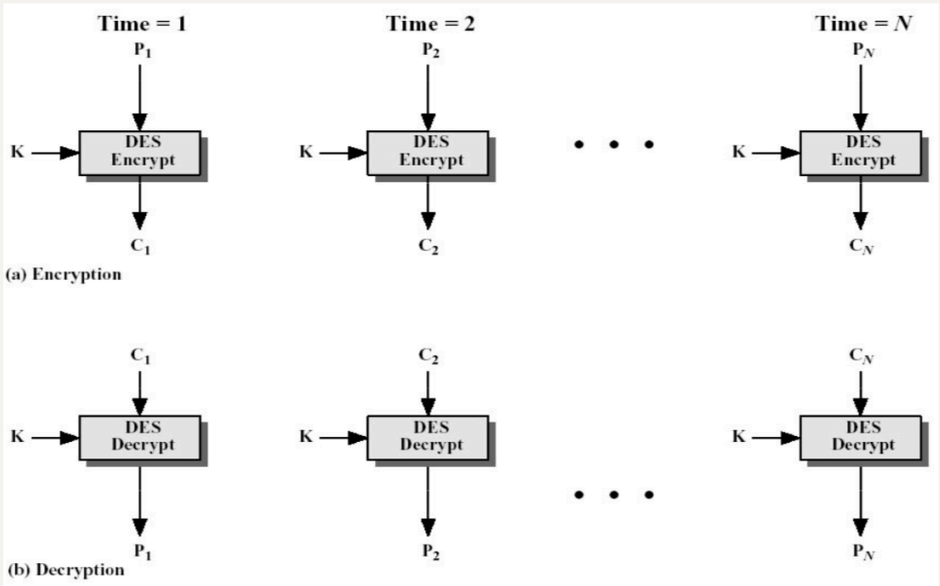
初始逆置换 IP^{-1}

和 ***初始置换IP*** 类似，有一个逆置换矩阵，用于把打散的数字恢复到它们 ***原来的位置***

DES的工作模式

 # 电子密码本 ECB

$$C_i = E_K(P_i) \Leftrightarrow P_i = D_K(C_i)$$



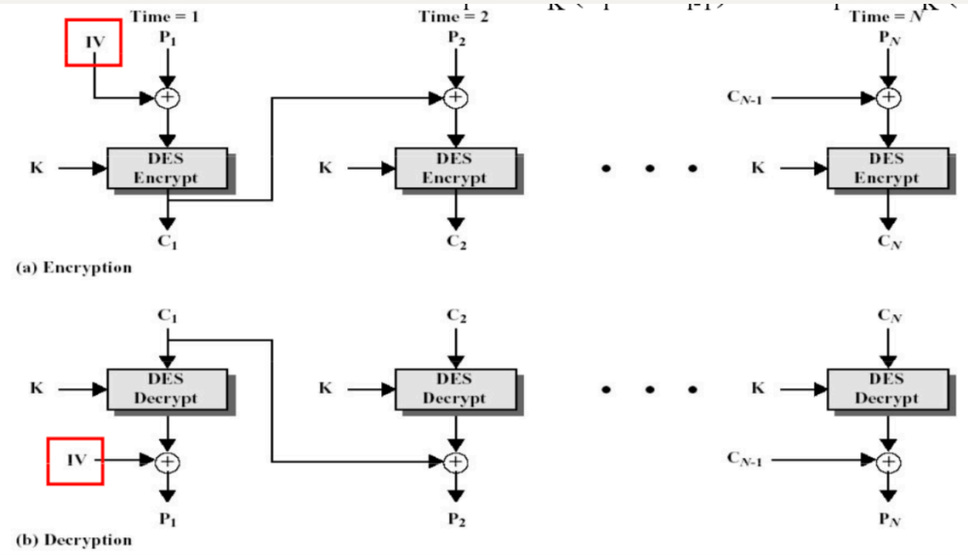
允许同一个分组密码密钥对对于一块的数据进行加密，并保证其安全性

特点：

- 简单有效
- 可以并行实现
- 相同明文生成相同的密文，同样的信息多次出现造成泄漏
- 对明文的主动攻击是可能的。信息快可以被替换、重排、删除、重放
- 误差传递：密文块损坏⇒仅对应的明文块损坏
- 适合于传输短信息

密码分组链接 CBC

$$C_i = E_K(P_i \oplus C_{i-1}) \Leftrightarrow P_i = D_K(C_i) \oplus C_{i-1}$$



前一轮加密的结果会和后一轮的明文进行异或⊕运算

特点：

- 没有已知的并行实现算法
- 相同的明文，往往会产生不同的密文，即隐藏明文的模式信息
- 需要有一个共同的初始化向量 IV
- 单个密文块的损坏，会导致解密过程中，两块明文的损坏
- 安全性好于ECB
- 适合于传输长度大于64位的报文，还可以进行用户鉴别，是大多数系统（SSL、IPSec）的标准

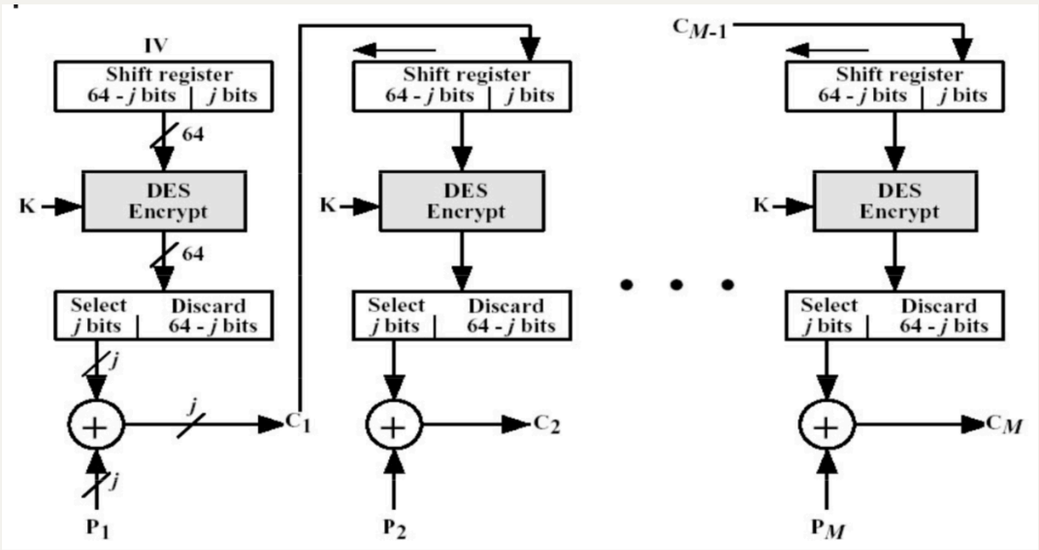
密码反馈 CFB

分组密码⇒流密码（像一个管道一样，流进去k个bit，流出k个bit）

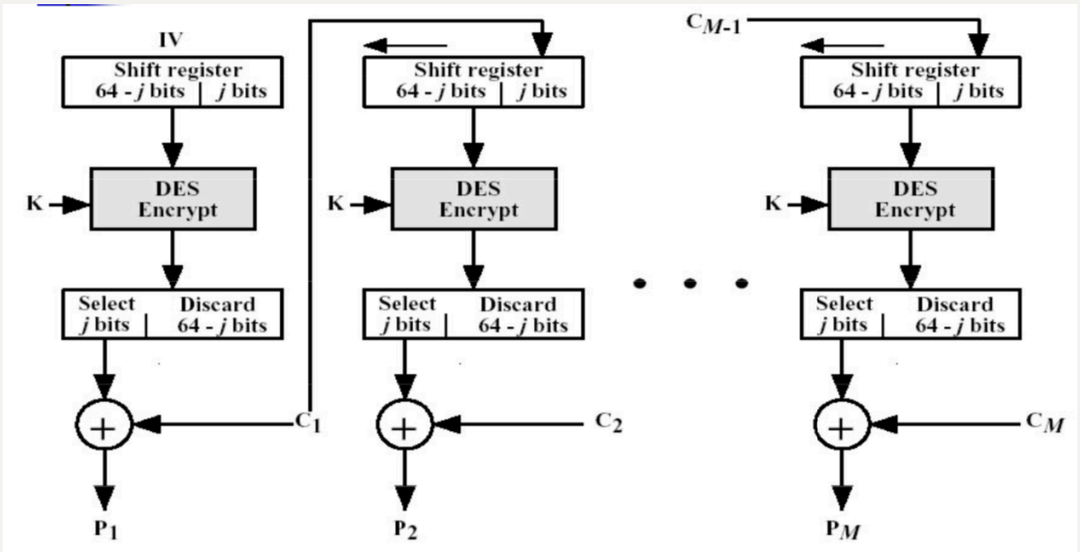
假定：S_i 为64位移位寄存器，传输单位为 $j\ bit$

加密：$C_i = P_i \oplus (E_K(S_i)$的高$j$位$)$，$S_{i+1} = (S_j << j \mid C_i)$

加密移位寄存器中的内容，而不是直接加密明文



解密：$P_i = C_i \oplus (E_K(S_i)$的高$j$位$)$，$S_{i+1} = (S_i << j \mid C_i)$



特点：

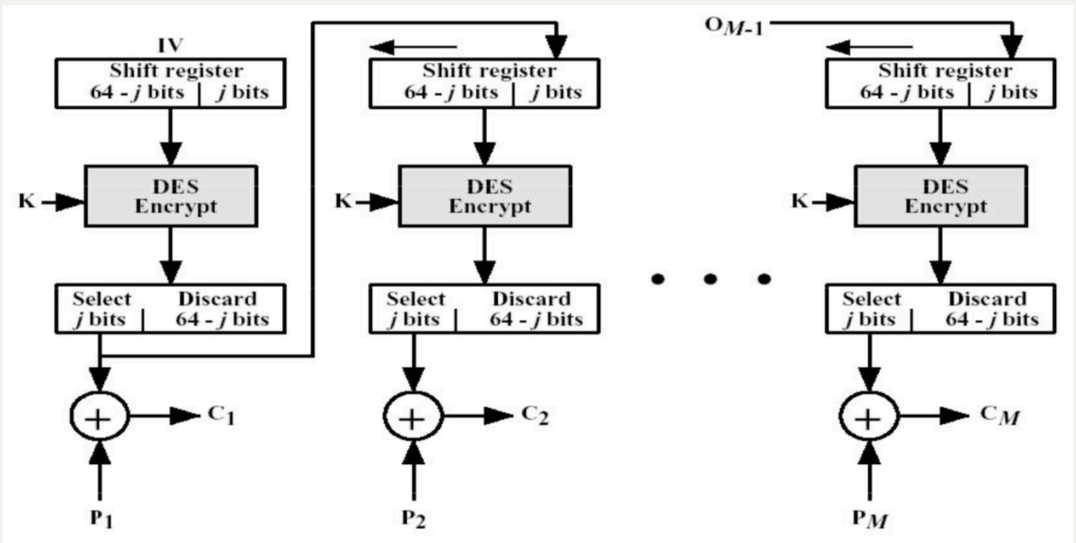
- 没有已知的并行实现算法
- 隐藏了明文信息，即相同的明文会得到不同的加密结果
- 需要共同的移位寄存器初始值 IV
- 对于不同的消息，IV 必须唯一
- 误差传递：一个单元损坏会影响多个单元（损坏指的是，接收端C_i损坏）

输出反馈 OFB

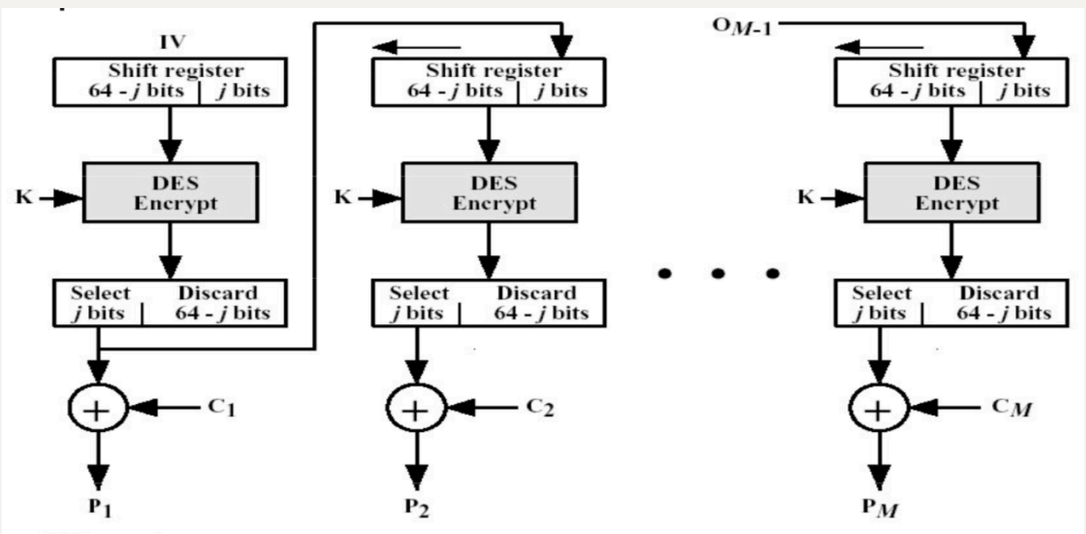
分组密码⇒流密码

假定：S_i 为64位移位寄存器，传输单位为 $j\ bit$

加密：$C_i = P_i \oplus (E_K(S_i)$的高$j$位$)$，$S_{i+1} = (S_j << j) \mid (E_K(S_i)$的高$j$位$)$



解密：$P_i = C_i \oplus (E_K(S_i)$的高$j$位$)$，$S_{i+1} = (S_i << j) \mid (E_K(S_i)$的高$j$位$)$



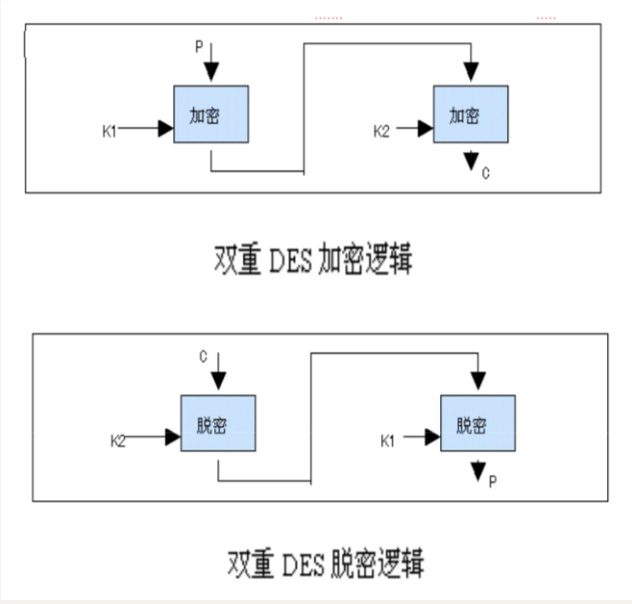
特点：

- 没有已知的并行实现算法
- 隐藏了明文模式
- 需要共同的移位寄存器初始值 IV

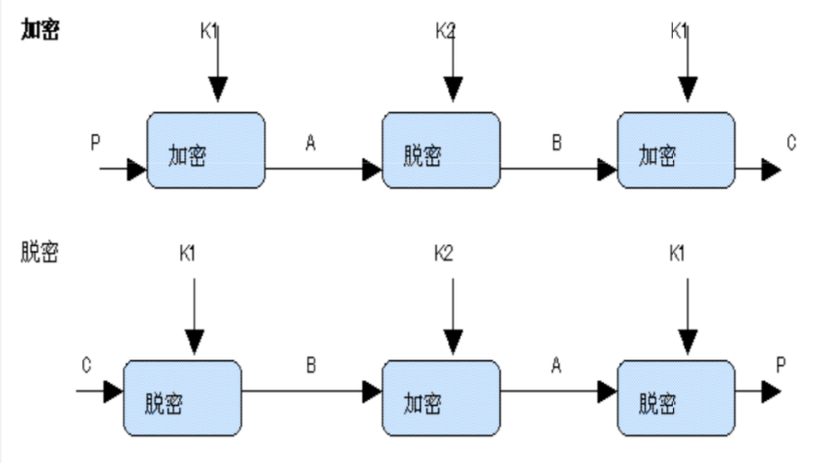
- 对于不同的消息，IV 必须唯一
- 误差传递：在解密端，一个密文C_i损坏只影响对应单元
- 对明文的主动攻击是可能的，信息块可能被替换、重排、删除、重放
- 安全性较CFB差

多重DES

两重DES



三重DES



> *为什么加密过程，不是 加密→加密→加密，而是 加密→脱密→加密?*

在计算机网络中，有的机器支持三重加密，而有的机器只支持单重加密

那么，假如第二步是加密，那么当单重加密和三重加密在网络中混合使用的时候，单重加密的机器就无法正常工作

然而，现在的第二步是脱密，对于单重加密的机器而言，它可以和三重加密的机器兼容