

密码学

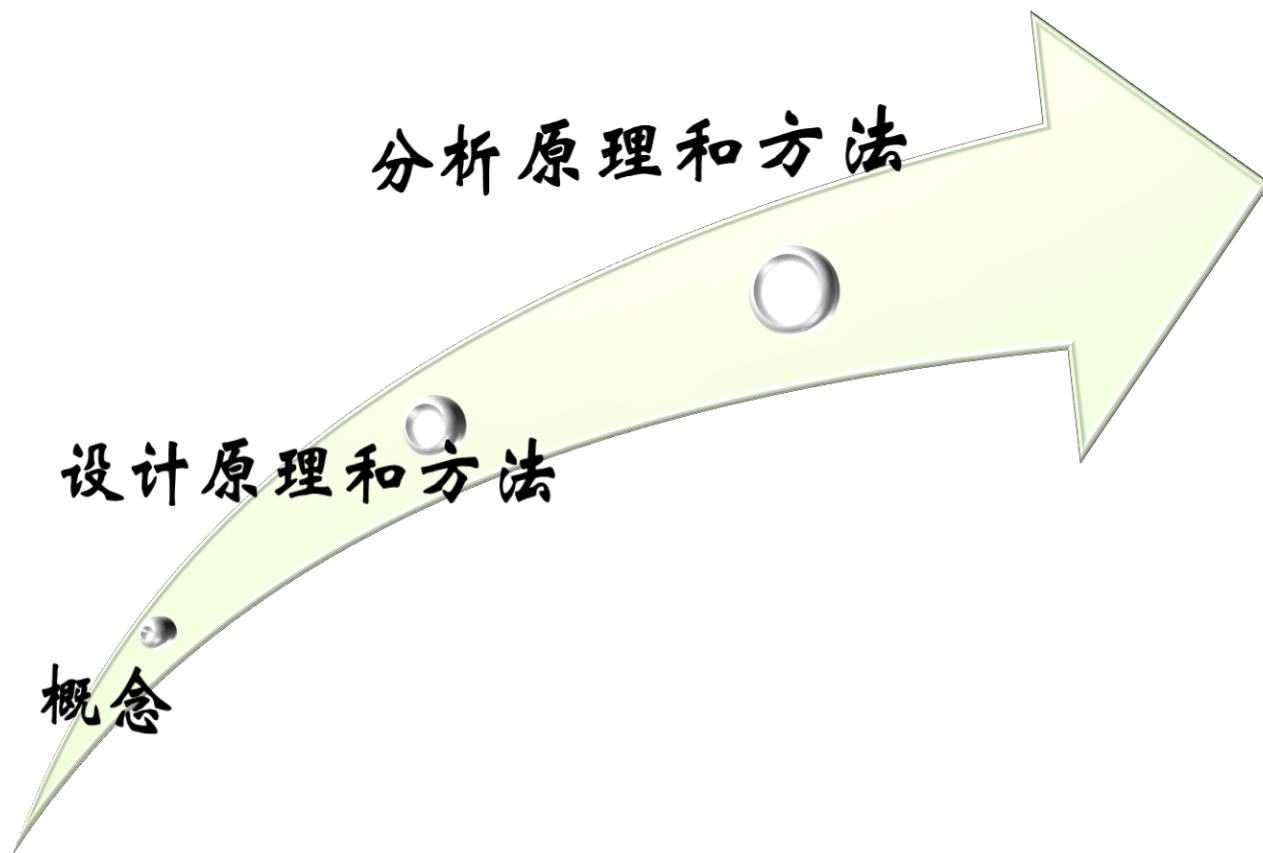
范明钰

信息安全研究中心

本部分主要内容

- 分组密码(DES、AES、Camellia)
- 序列密码
- Hash函数
- 公开密码(RSA、Elgamal、ECC)

基本线索



现代密码的特征

现代
密码
特征

处理的信息
不再是字符

实现算法的
手段
不再是手工或
机械

算法分析手
段
日趋成熟

约束条件--KERCKHOFF原则

1883年
Kerchoffs第一次明确提出密码编码的原则：加密算法应建立在C1、C2、C3的基础上

这一原则被普遍认同，成为衡量密码强度的标准，实际上也成为古典密码和现代密码的分界线

- C1：密码算法公开
- C2：敌方拥有大量数量的密文
- C3：敌方拥有一定数量的明密对应

分组密码

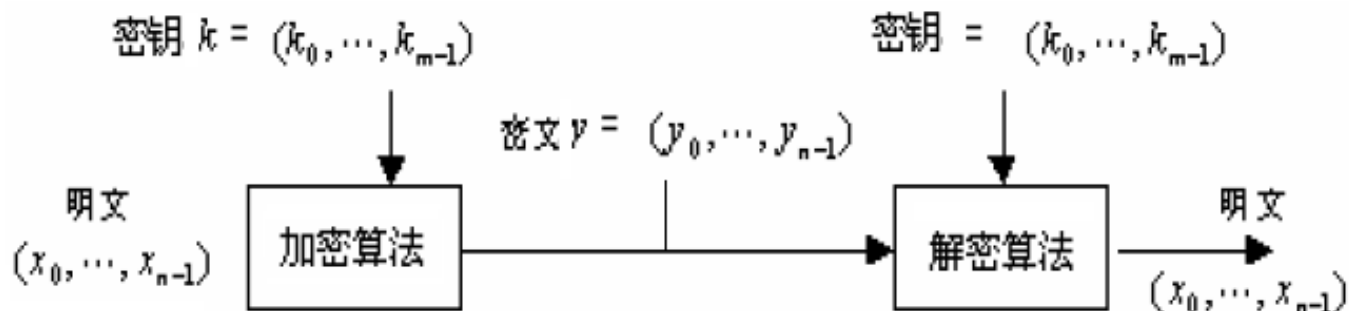
- ◆ 定义
- ◆ 分组密码的一般设计原理
- ◆ 分组密码的典型攻击方法

现代常规分组加密算法： 举例

- DES
- Triple DES
- IDEA
- Blowfish
- CAST-128
- Camellia
- AES

分组密码定义

- 划分成长度等长的分组进行处理
- 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序列



分组密码模型

分组密码的表述

■ 设 K 为密钥空间， n 为分组长度，则分组密码加密变换和解密变换如下：

■ $E = \{E_k \mid E_k: F_2^n \rightarrow F_2^n \text{ 是一一映射}, k \in K\}$


■ $D = \{D_k \mid D_k: F_2^n \rightarrow F_2^n \text{ 是一一映射, 且 } D_k \text{ 是 } E_k \text{ 的逆}, k \in K\}$

■ 若不区别 F_2^n 中点 $X = (x_0, x_1, \dots, x_{n-1})$ 和其所对应的二进制数，

$$\|X\| \triangleq (x_0 x_1 \cdots x_{n-1})_2$$

■ 则每个 E_k 或 D_k ($k \in K$) 均等同一个 2^n -置换。

分组密码的本质：集合中的代替



在密钥的控制之下，
从一个足够大而且
足够好的代替子集
中，简单而迅速地
选取出一个代替

分组密码的设计原则

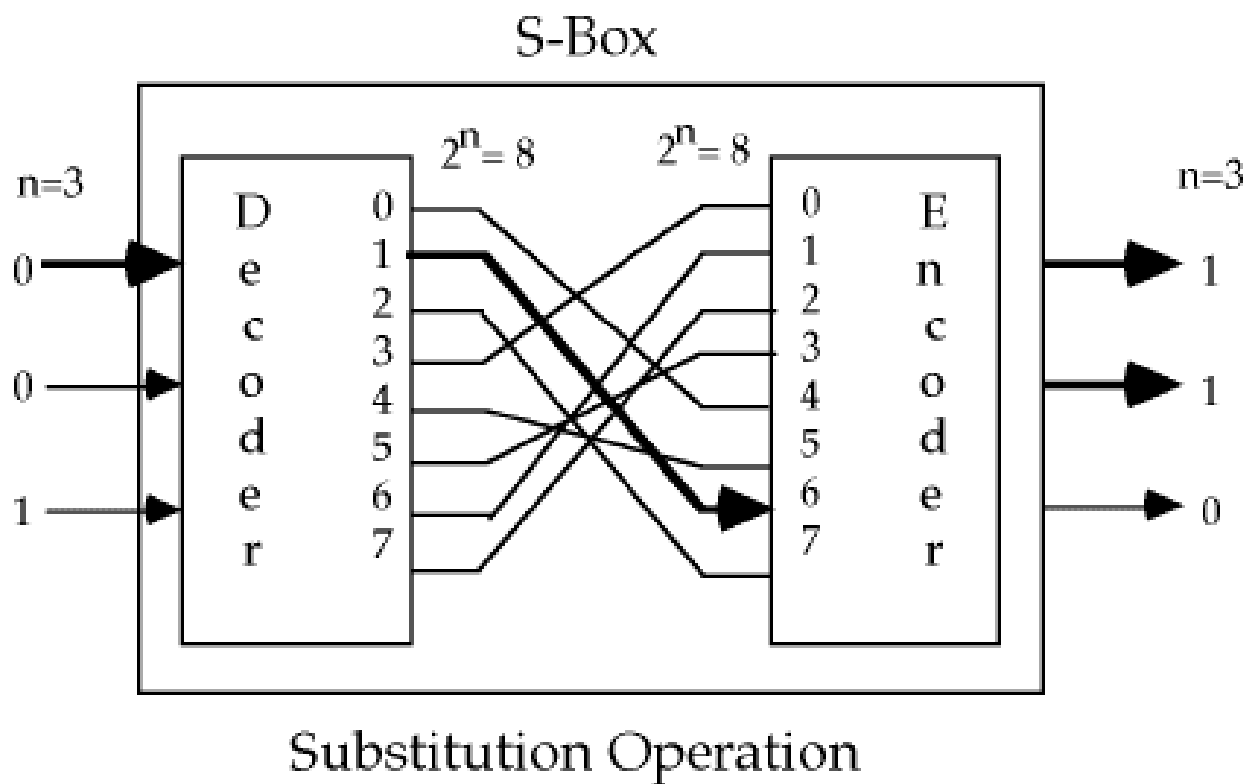
- 分组长度 n : 足够大, 防止对明文的穷搜攻击奏效
- 密钥空间 K : 足够大, 防止对密钥的穷搜攻击奏效
- 要使密文和明文以及密文和密钥之间的依赖关系相当复杂, 使密码分析者无法利用其依赖性 → 混乱
- 细节上: 使每bit密钥影响一半以上bit密文, 以防止对密钥进行逐段破译; 每bit明文也应影响一半以上bit密文, 以便隐蔽明文的统计特性 → 扩散
- 部分密钥被破译后, 分组密码仍有一定的抗攻击能力
→ 稳定的安全性

分组密码的设计技术手段

- 1949 Claude Shannon 提出S-P 网络
- 基于密码学的两个基本操作
 - *substitution* (S-box)
 - *permutation* (P-box)

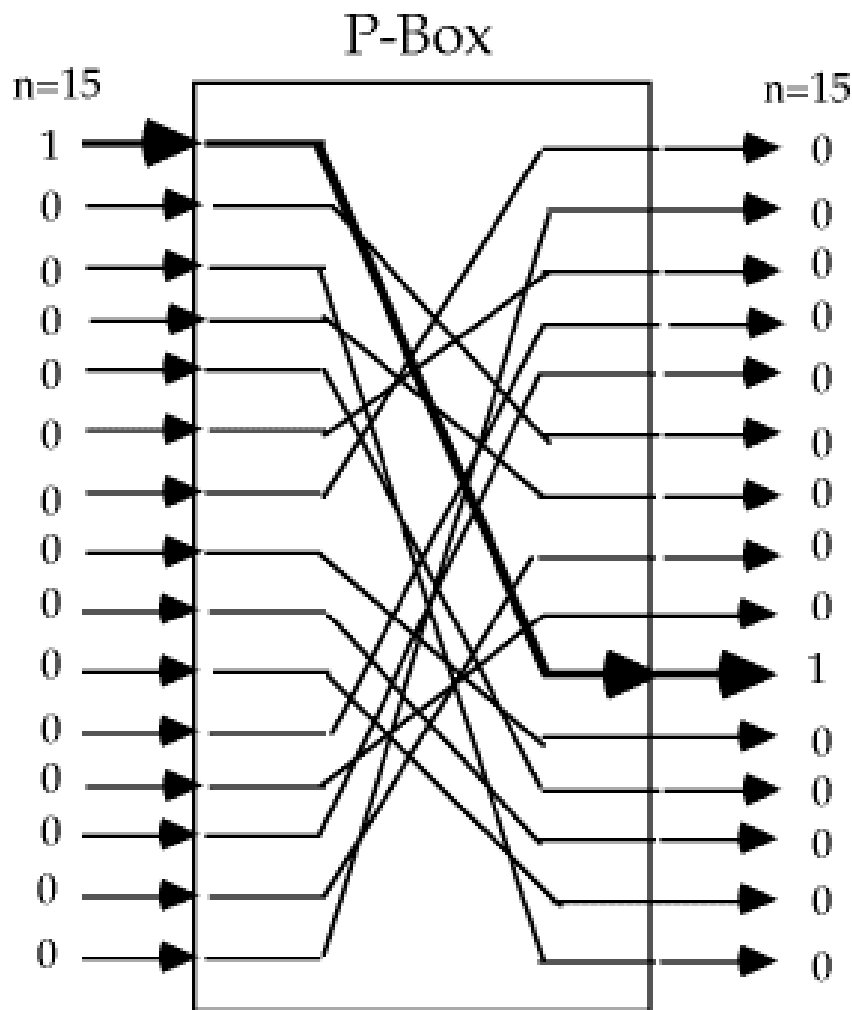
代替运算（变换）：S-Box

- 一个二进制字（分组）用其它二进制字（分组）替换
- 是一个大的查表运算



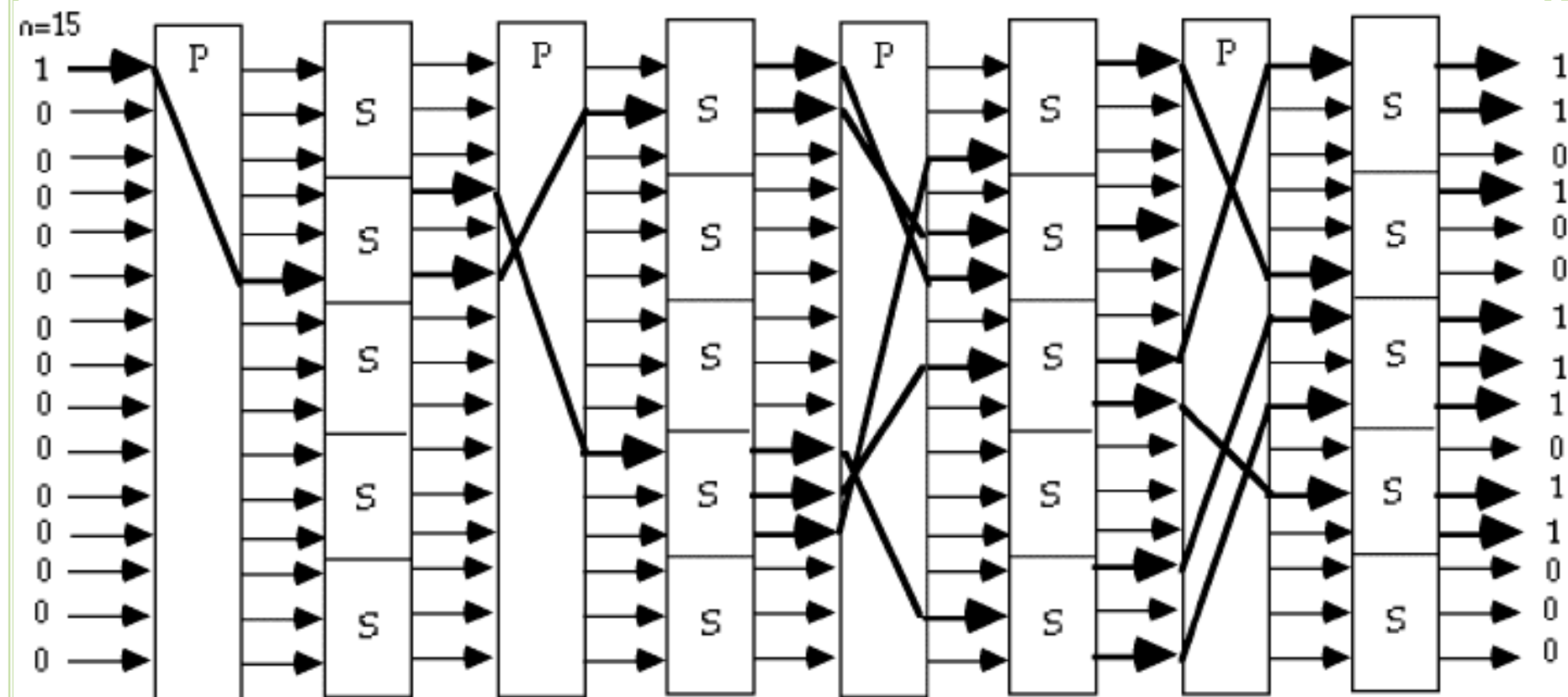
置换运算（变换）：P-BOX

- 二进制字（分组）的排列次序被打乱
- 是重新排序



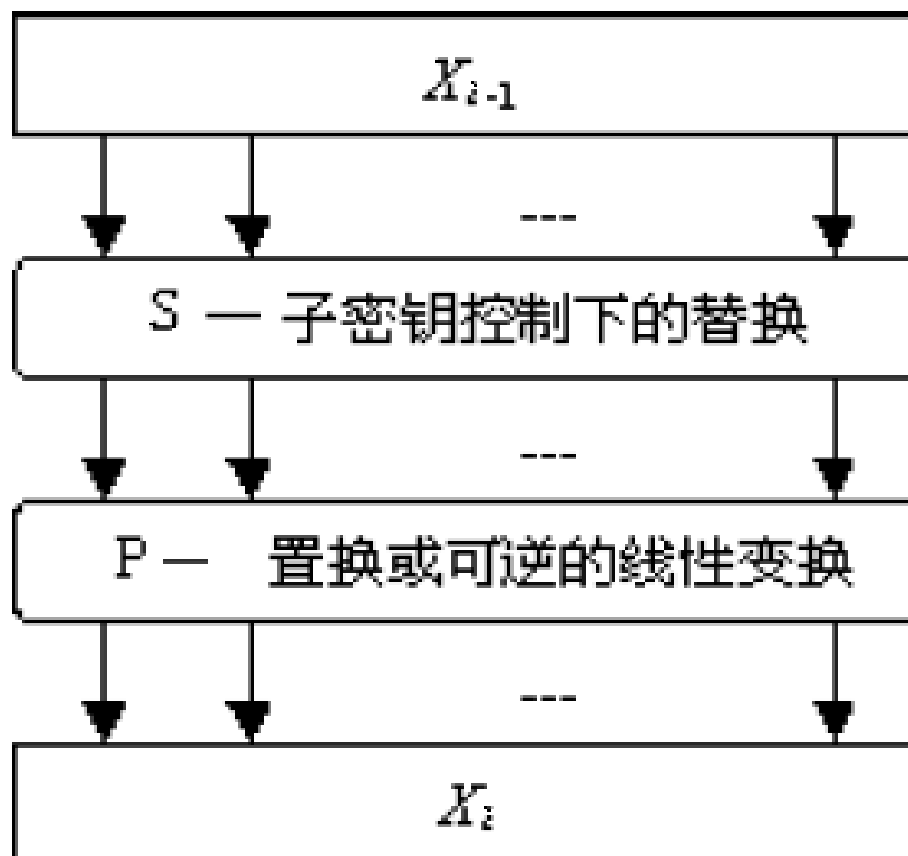
Permutation or Transposition Function

混合变换



Substitution-Permutation Network, with the Avalanche Characteristic

加上密钥控制



一轮 SP 网络加密过程

基本原理：采用乘积密码

S-Boxes

- 混合作用 (confusion)

P-Boxes

- 扩散作用 (diffusion)

实际使用的考虑

有两种方法

- **分别定义**每个替换、置换的逆：复杂度增加
- 定义一种**结构**，容易求逆：可以使用基本的相同编码或硬件，用于加密和解密。例如Feistel结构

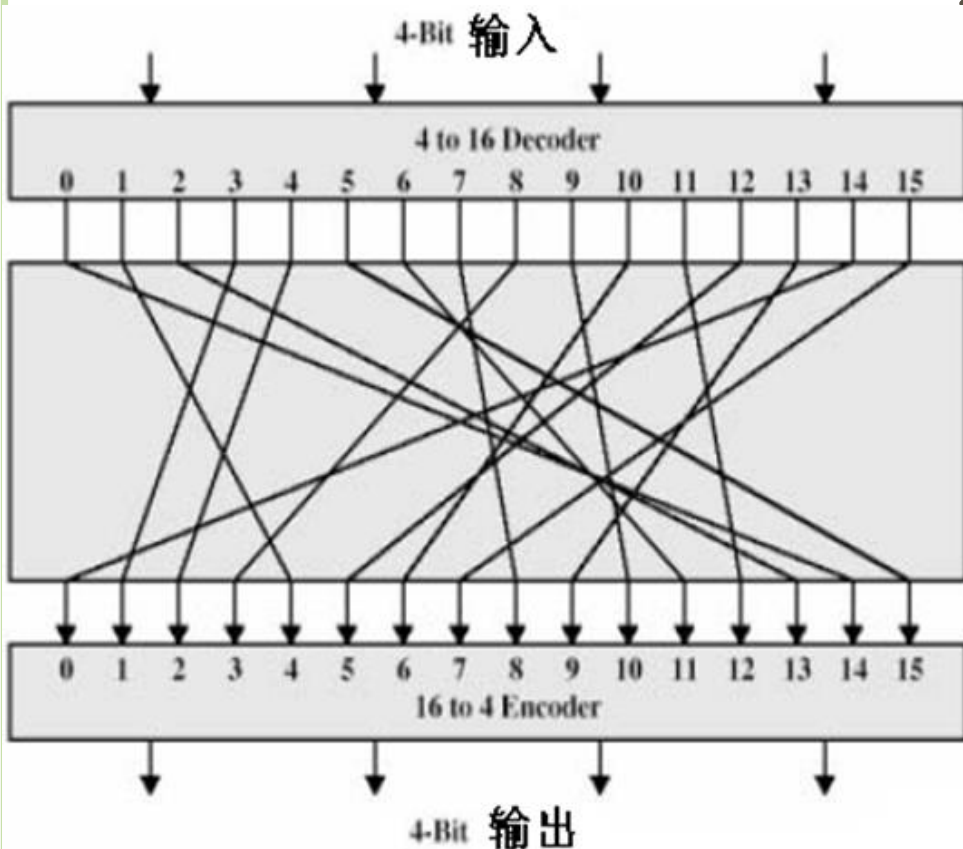
FEISTEL 密码

Horst Feistel
(working at IBM
Thomas J Watson
Research Labs) 70
年代初设计

基本思想：把输入
块分成左右两部分，
 $L(i-1)$ 和 $R(i-1)$ 。变换
时在第 i 轮只使用
 $R(i-1)$

函数 g ：每个阶段 i 内
 g 的工作，由第 i 个
密钥（称为子密钥）
控制

FEISTEL结构设计之动机



一般nbit-nbit分组代替($n=4$)

解决了下述问题

一一映射

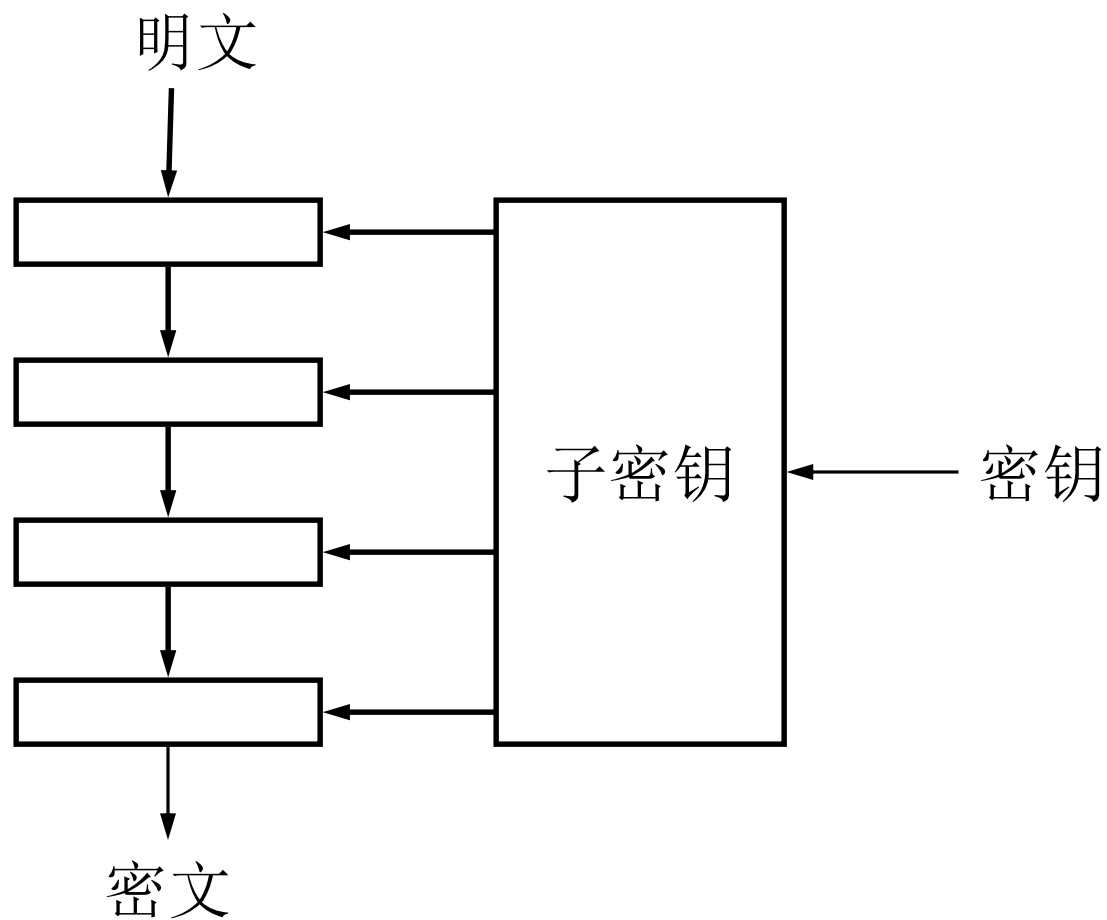
当分组 n 较小时，
等价于代替变换

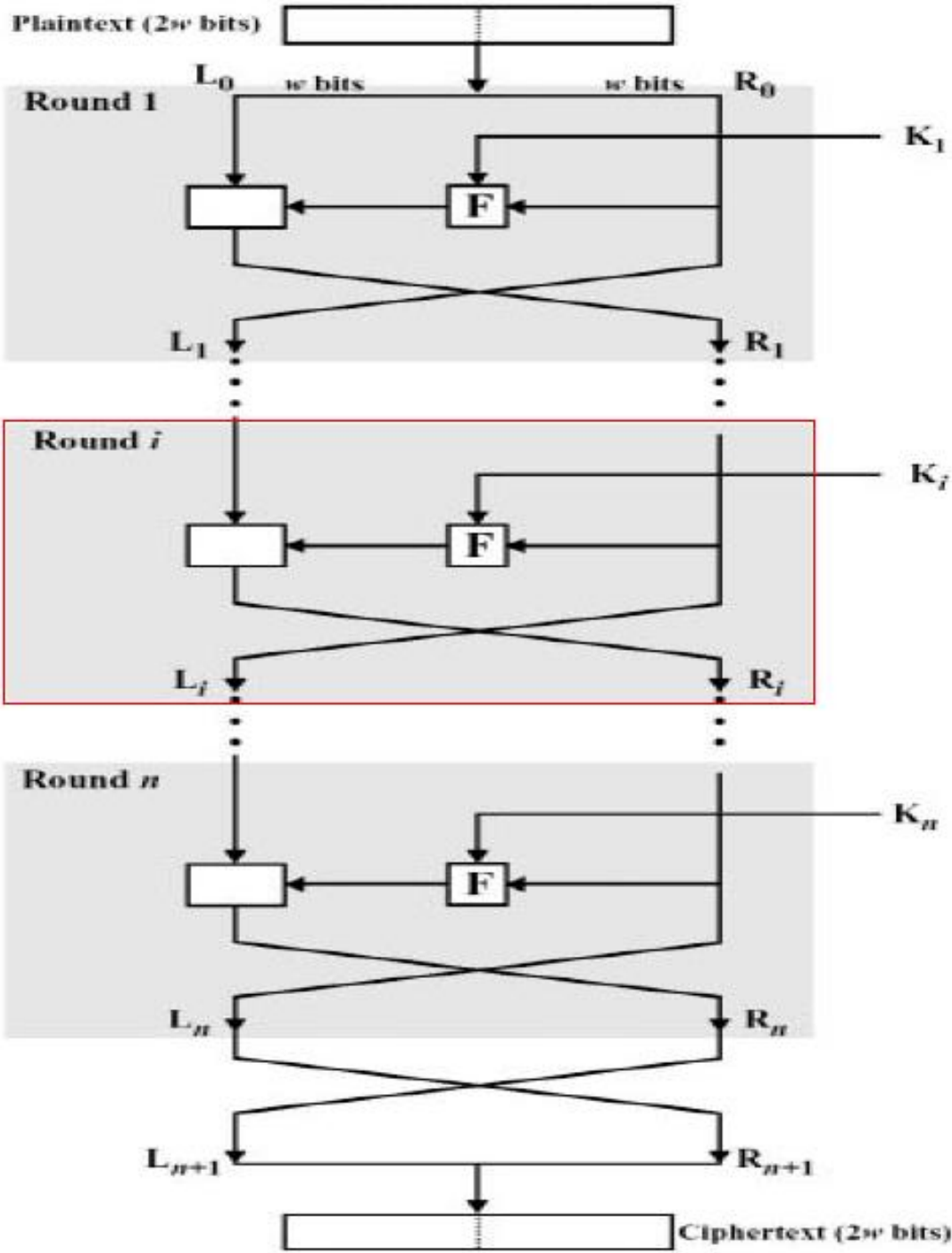
当分组 n 较大时，如
 $n=64$ ，无法表达任意
变换

范明

王本利

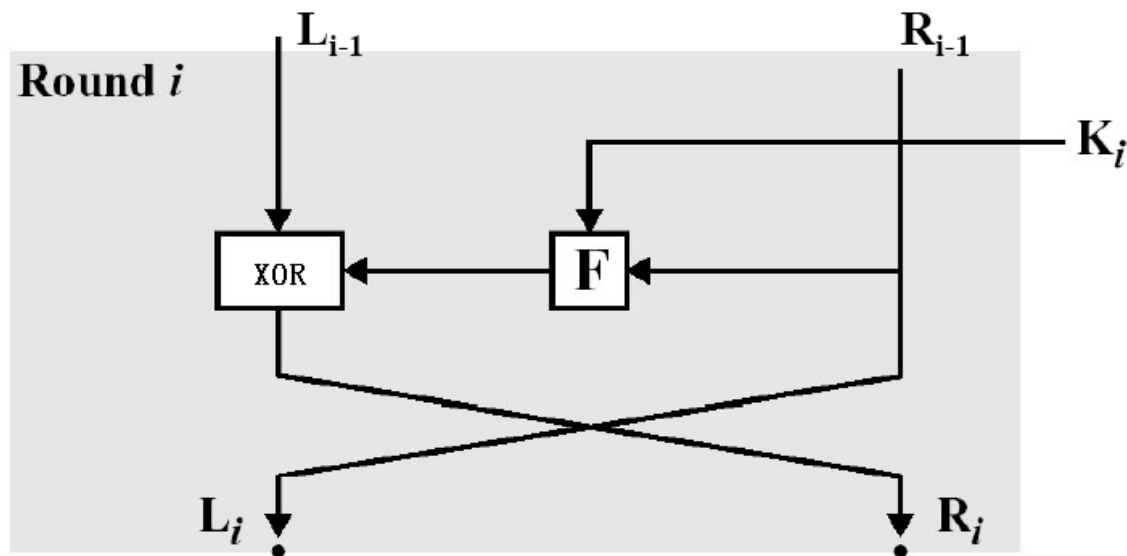
多轮循环思想





FEISTEL 结构图

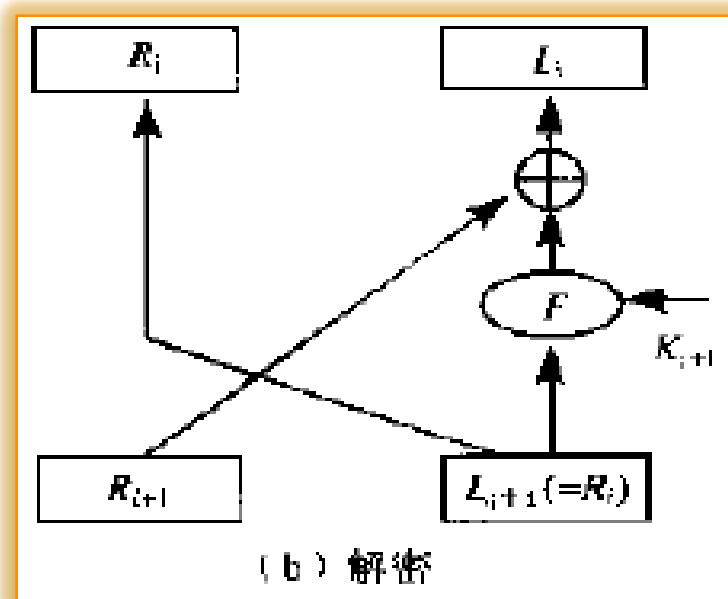
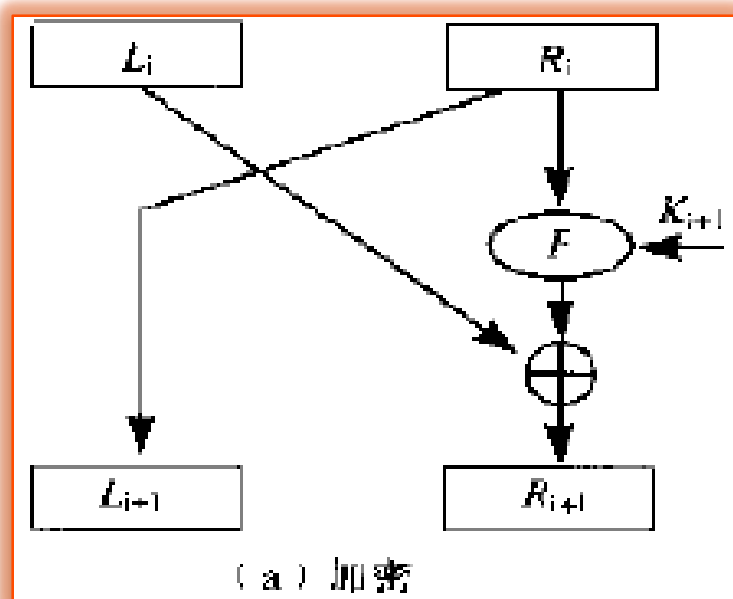
FEISTEL结构轮细节



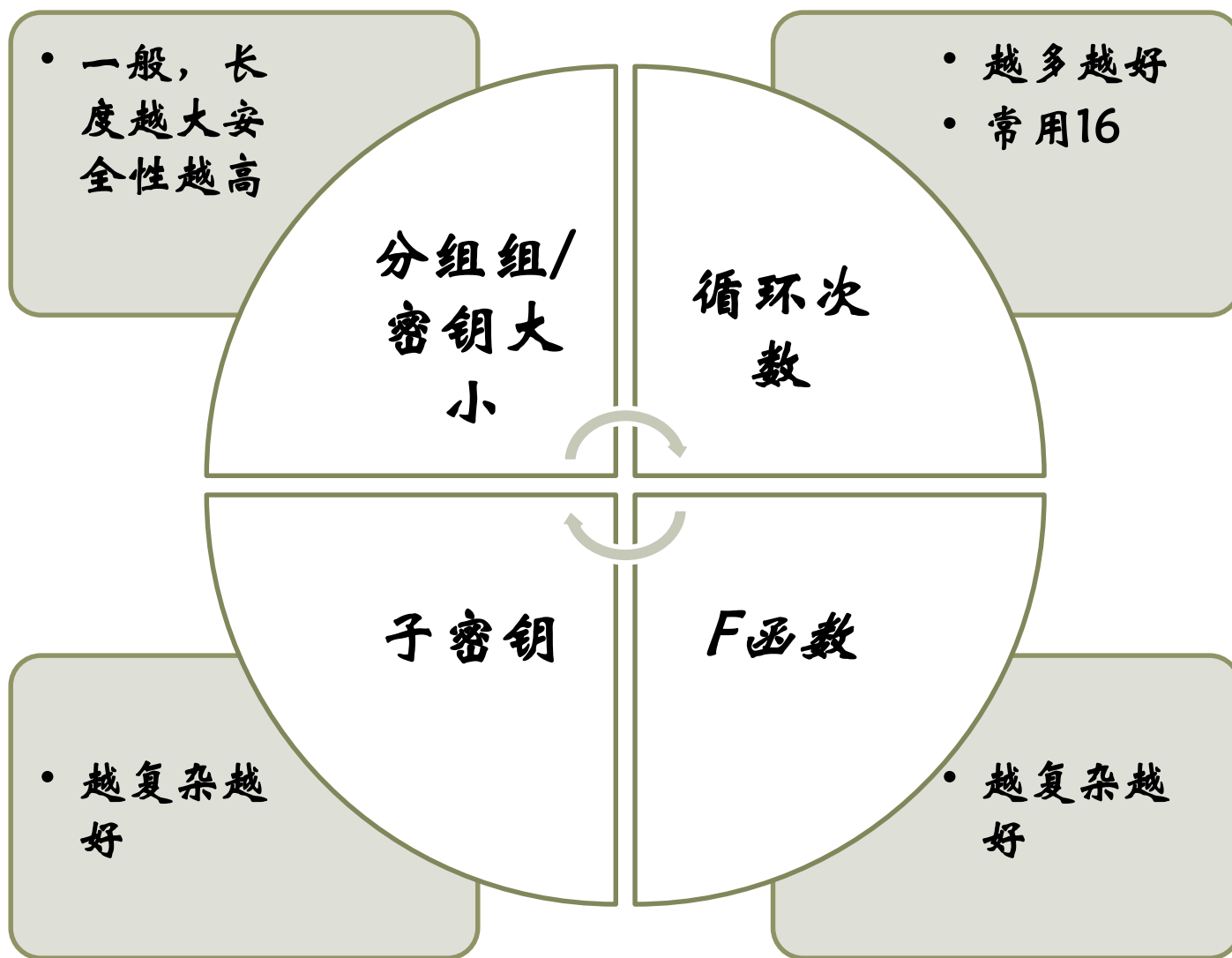
加密: $L_i = R_{i-1}$; $R_i = L_{i-1} \oplus F(R_{i-1}, K_i)$

解密: $R_{i-1} = L_i$; $L_{i-1} = R_i \oplus F(R_{i-1}, K_i) = R_i \oplus F(L_i, K_i)$

单轮的加密与解密



FEISTEL密码的密钥参数

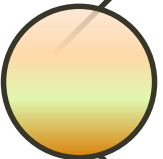


应用



Feistel结构把F函数转化为一个置换。

许多分组密码采用Feistel结构：
FEAL, GOST, LOKI, E2,
Blowfish, Camellia和RC5等



一般，每一轮中的F函数都相同，仅密钥不同。

可推广为每一轮中的F函数也变化，如Khufu和MD4等算法

应用举例：DES

■ *D*ata *E*ncryption *S*tandard

- Adopted in 1977 by NBS (now NIST) as FIPS PUB 46
- 64-bit blocks with 56-bit key (plus 8 parity bits for a total of 64)
- Same steps and same key complete the decryption

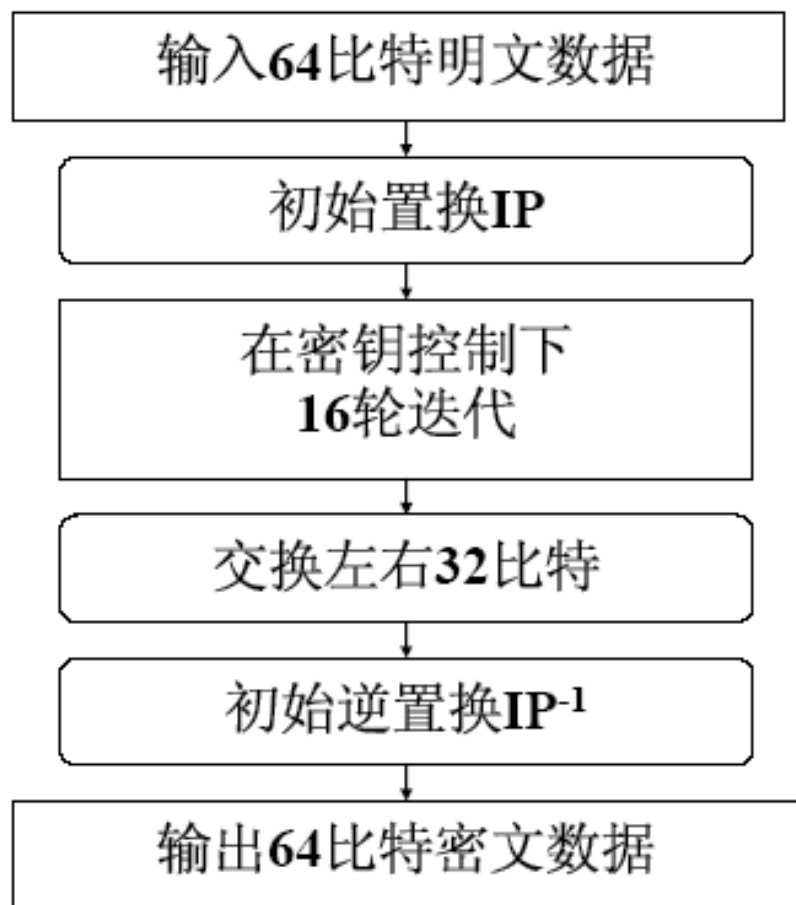
■ 16 rounds

■ *F*: *E*-table + *S*-box + *P*-table (expansion/permutation + substitution/choice + permutation)

数据加密标准DES概述

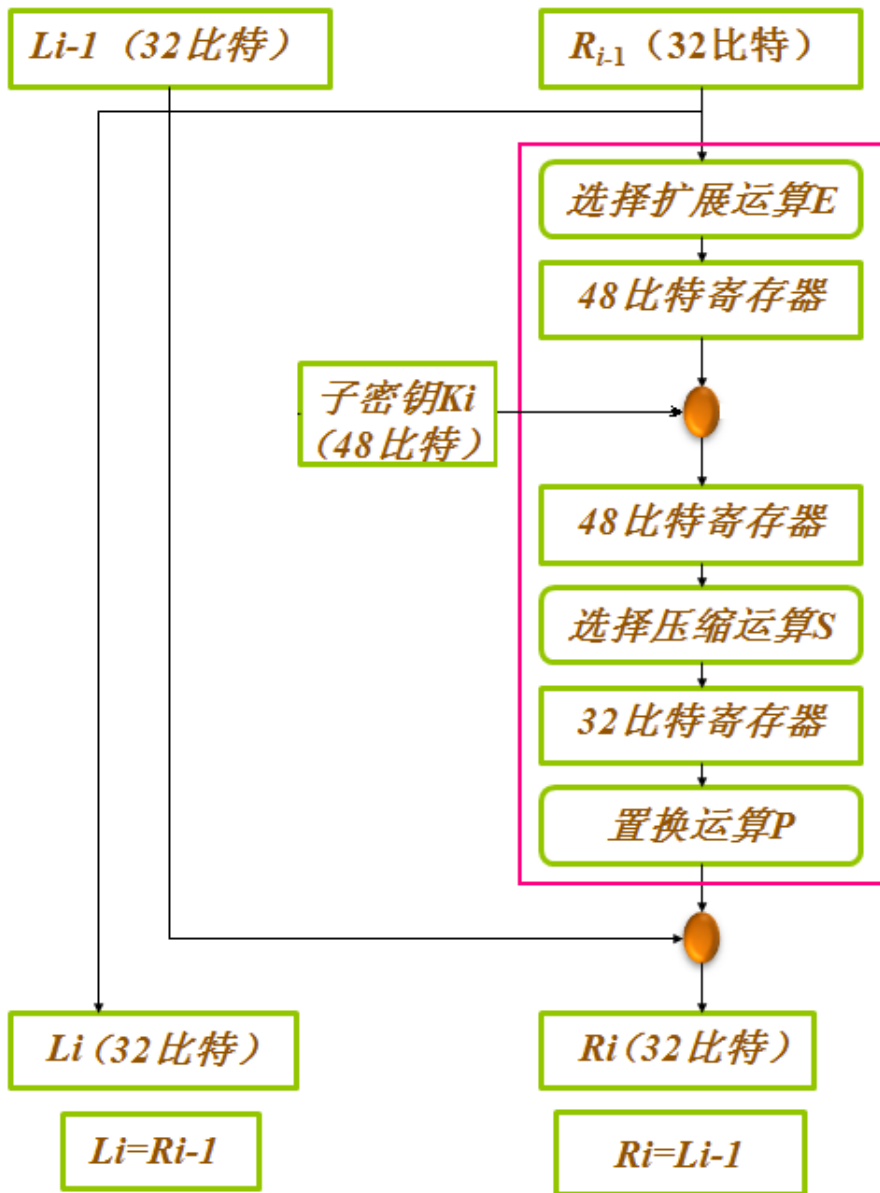
- IBM 设计出Lucifer密码(1971)
 - 由Horst Feistel带领的团队
 - 用128比特密钥加密64比特数据分组
- Tuchman-Mayer 牵头开发商业密码
 - 适合于单芯片实现
 - 密钥长度56比特，抗密码分析能力更强
 - 美国国家安全局介入
- 1973年美国国家标准局征求国家密码标准方案，IBM将其方案提交，并被采用，称为数据加密标准 (DES)

DES总体设计框图

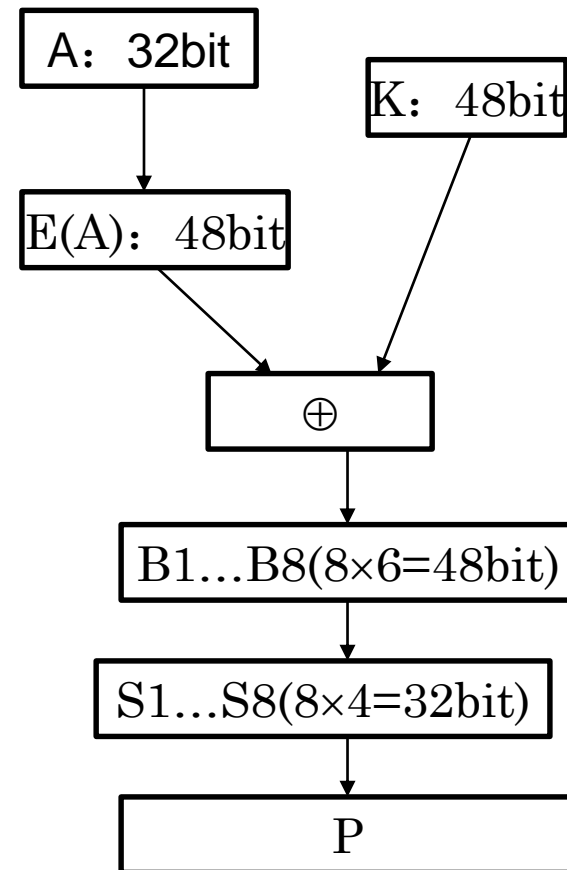


DES细节

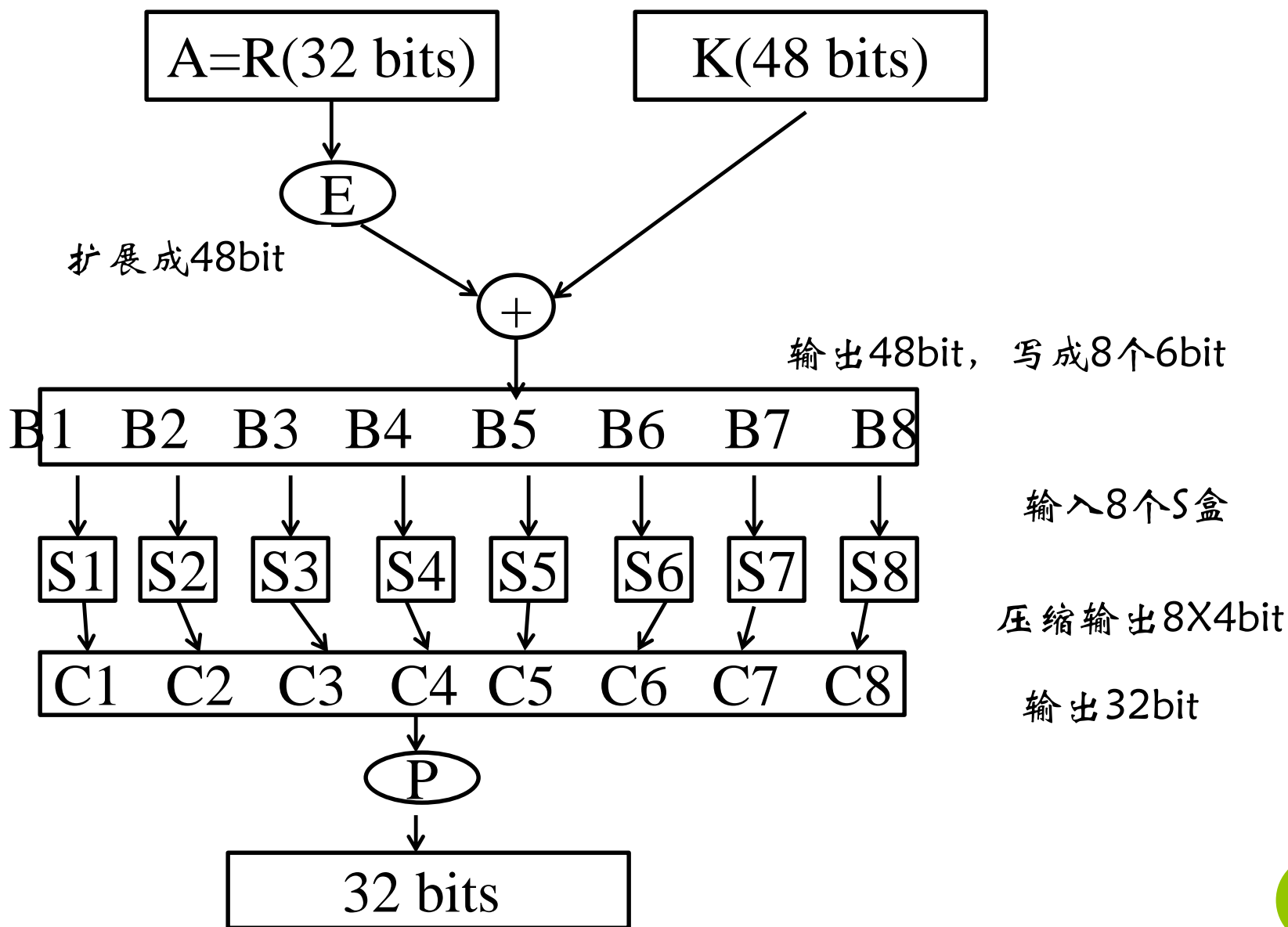
F的计算



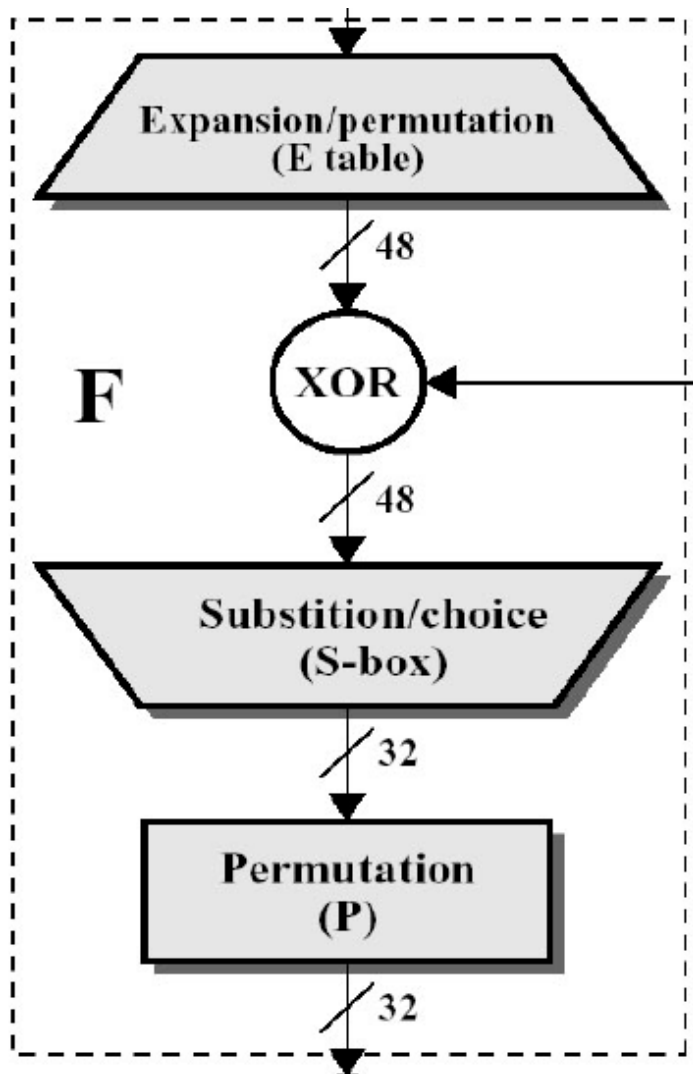
DES的一轮迭代



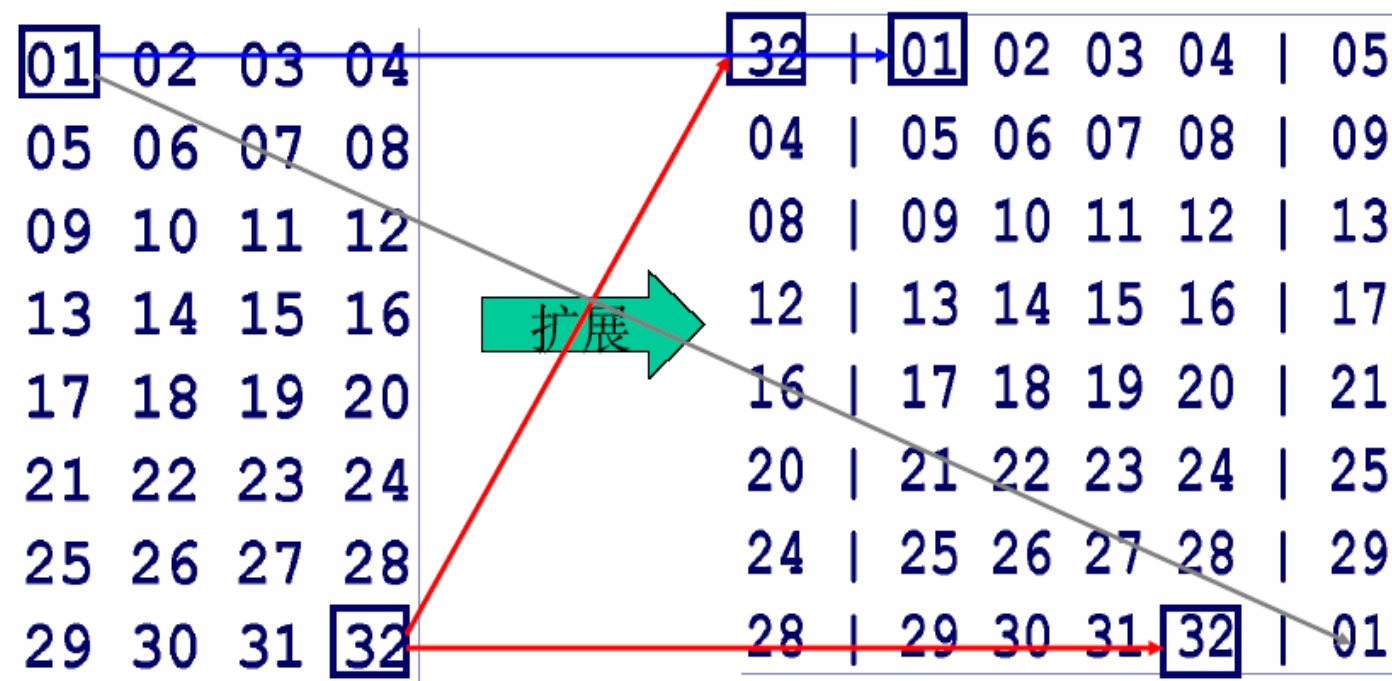
F函数细节



F函数总体框图



扩展E盒 —— 32位扩展到48位



- 起扩散作用
- 某一个比特的影响，几轮操作后会扩散到整个分组64位

课堂作业

- 将下述32bit数据按照DES的E盒扩展为48bit
- 1010 0111 0011 0010 0101 0010 0111 0100

压缩S-BOX：6入4出（BIT）

S₁

14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

S₂

15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9

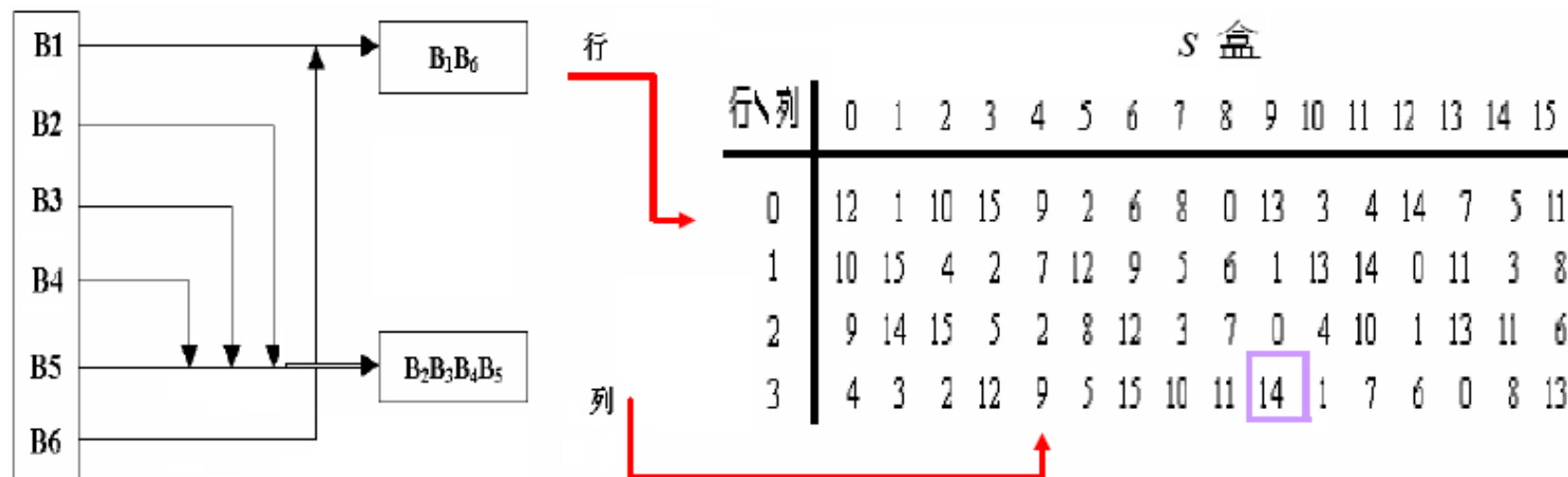
S₃

10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1
13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7
1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12

S₄

7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15
13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9
10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4
3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14

S-Box使用方法举例



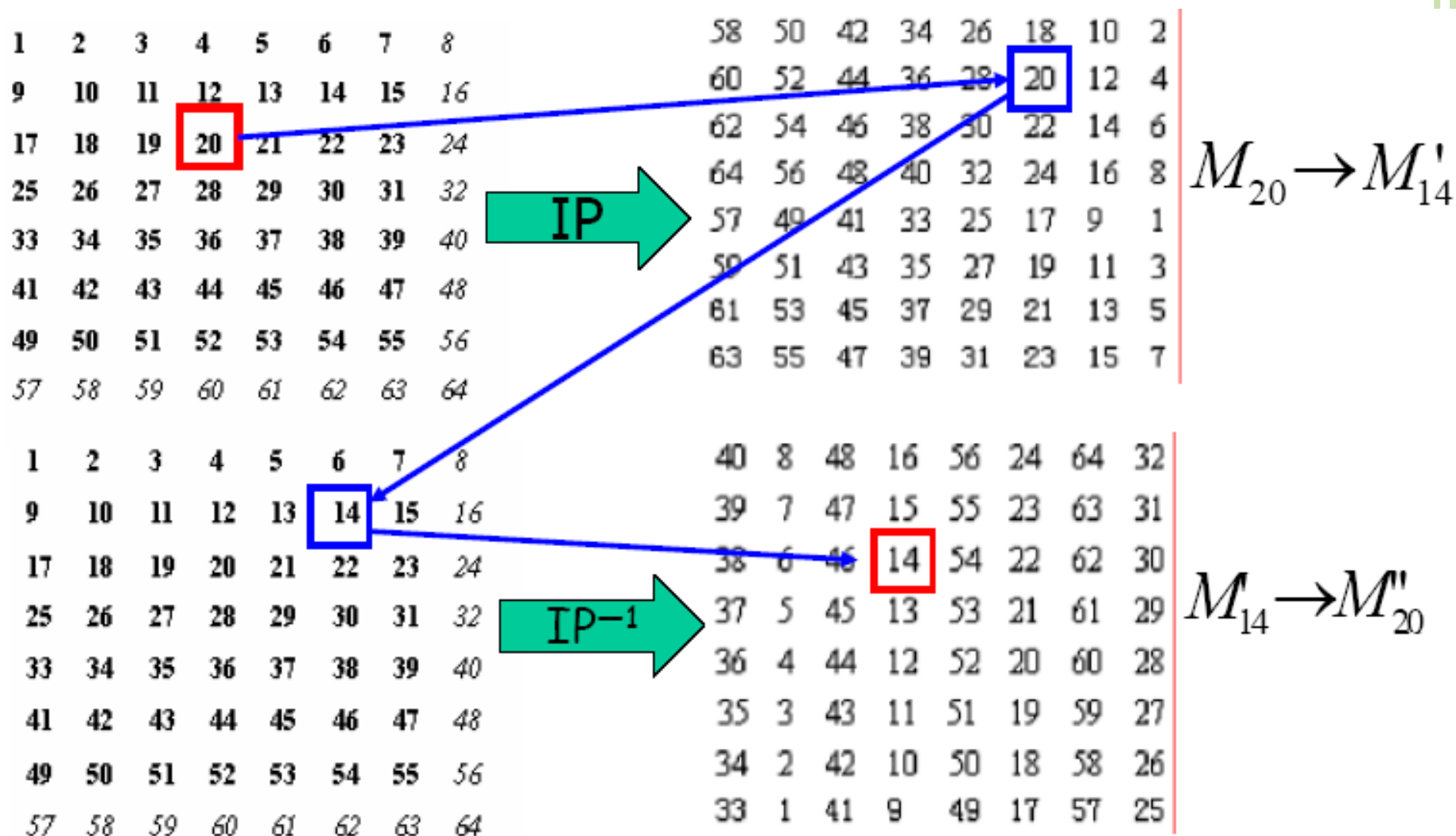
$$\begin{array}{c} b_1 b_2 b_3 b_4 b_5 b_6 \\ \hline 110011 \end{array} \Rightarrow \begin{array}{l} \text{行: } b_1 b_6 = 11_2 = 3 \\ \text{列: } b_2 b_3 b_4 b_5 = 1001_2 = 9 \end{array} \Rightarrow \begin{array}{c} \text{s 盒} \quad 3\text{行}9\text{列} \\ \hline \text{值: } 14 = 1110 \end{array}$$

$S_1(010011)$ 的值:

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
2	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

$$S_1(010011) = 0110$$

初始置换IP和IP-1



轮密钥的产生

- 输入64位密钥，使用56位
- 置换选择PC1将64位原始密钥置换输出56位
- 注意：若用大写英文字母作为密钥，常用它们的ASCII码作为二进制密钥。这是危险的，因为它们的ASCII码最高位均为0，而DES舍弃的是最低位！
- 56位密钥分为两组，每轮迭代分别循环左移1位或2位，作为下一轮的密钥输入
- 移位产生的值经置换选择2，输出48位作为轮函数子密钥

轮密钥生成总体框架

使用置换(PC-1)从64位输入密钥中选出56位的密钥，剩下的8位要么直接丢弃，要么作为奇偶校验位

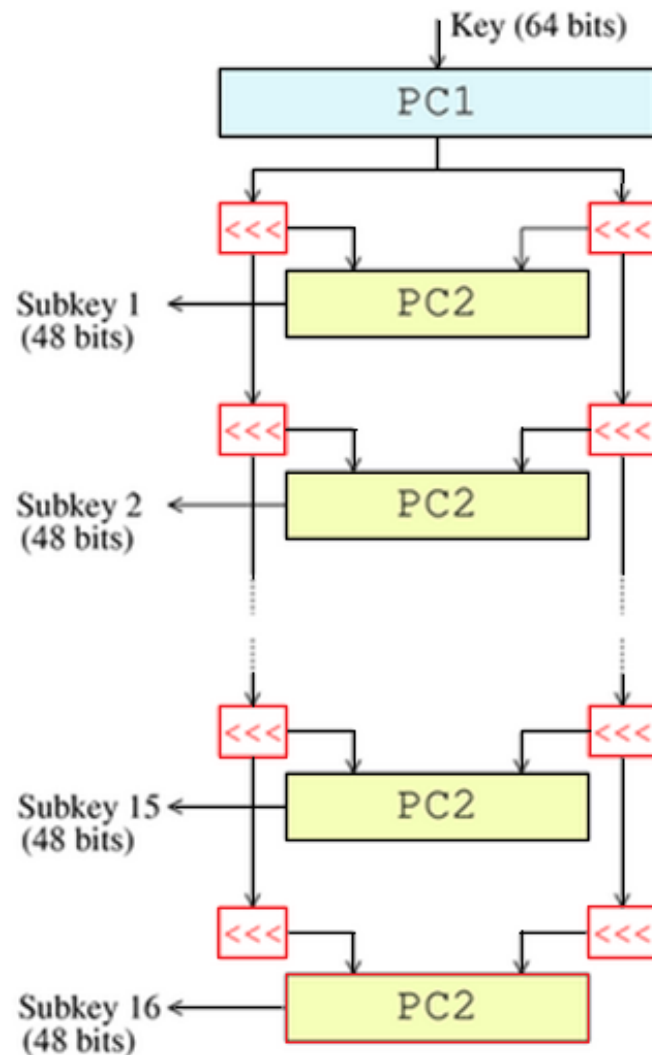
56位分成两个28位的半密钥；每个半密钥分别处理

每轮中，两个半密钥都左移1或2位（由轮次决定）

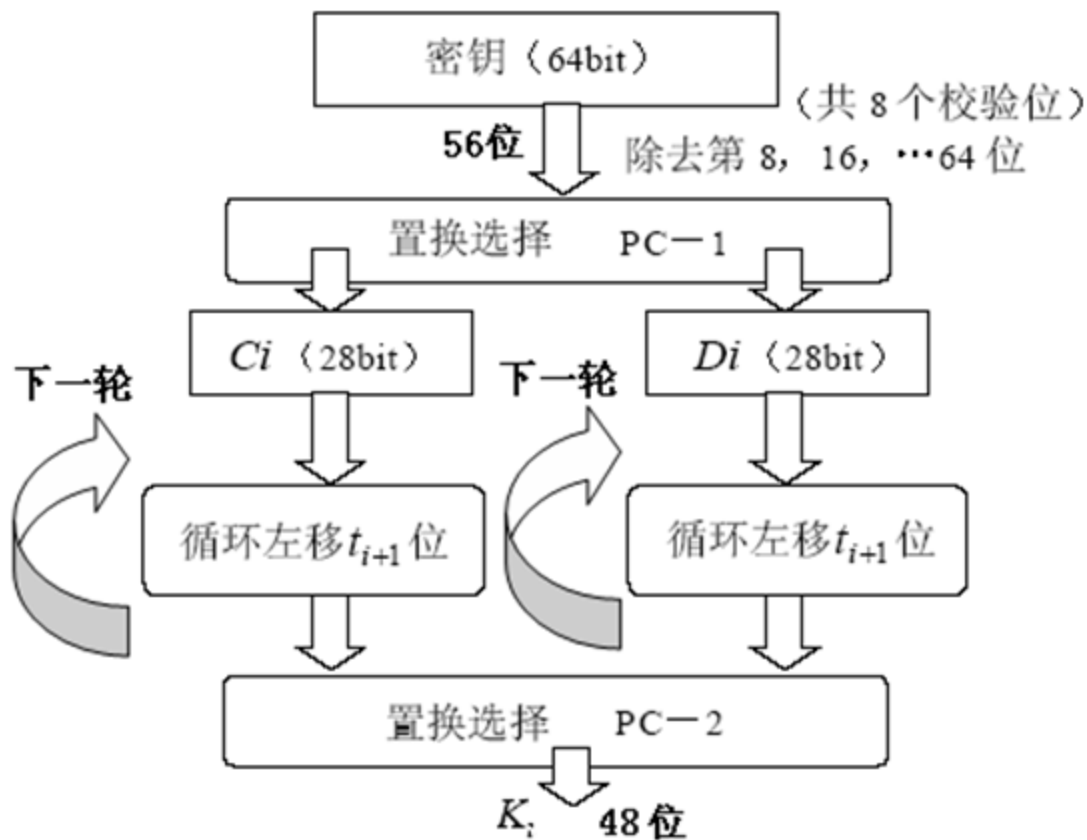
用置换(PC-2)产生48位子密钥——每个半密钥24位

移位（由<<标示）表明每个子密钥中使用了不同的位（每个位大致在16个子密钥中的14个出现）

解密时，除了子密钥输出的顺序相反外，密钥编排程的过程与加密完全相同



各轮子密钥的生成



PC-1

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

PC-2

14	17	11	24	1	5
2	28	15	6	21	10
23	19	12	4	26	8
16	7	27	20	13	2
41	52	31	37	47	55
30	40	51	45	33	48
44	49	39	56	34	53
46	42	50	36	29	32

移位次数

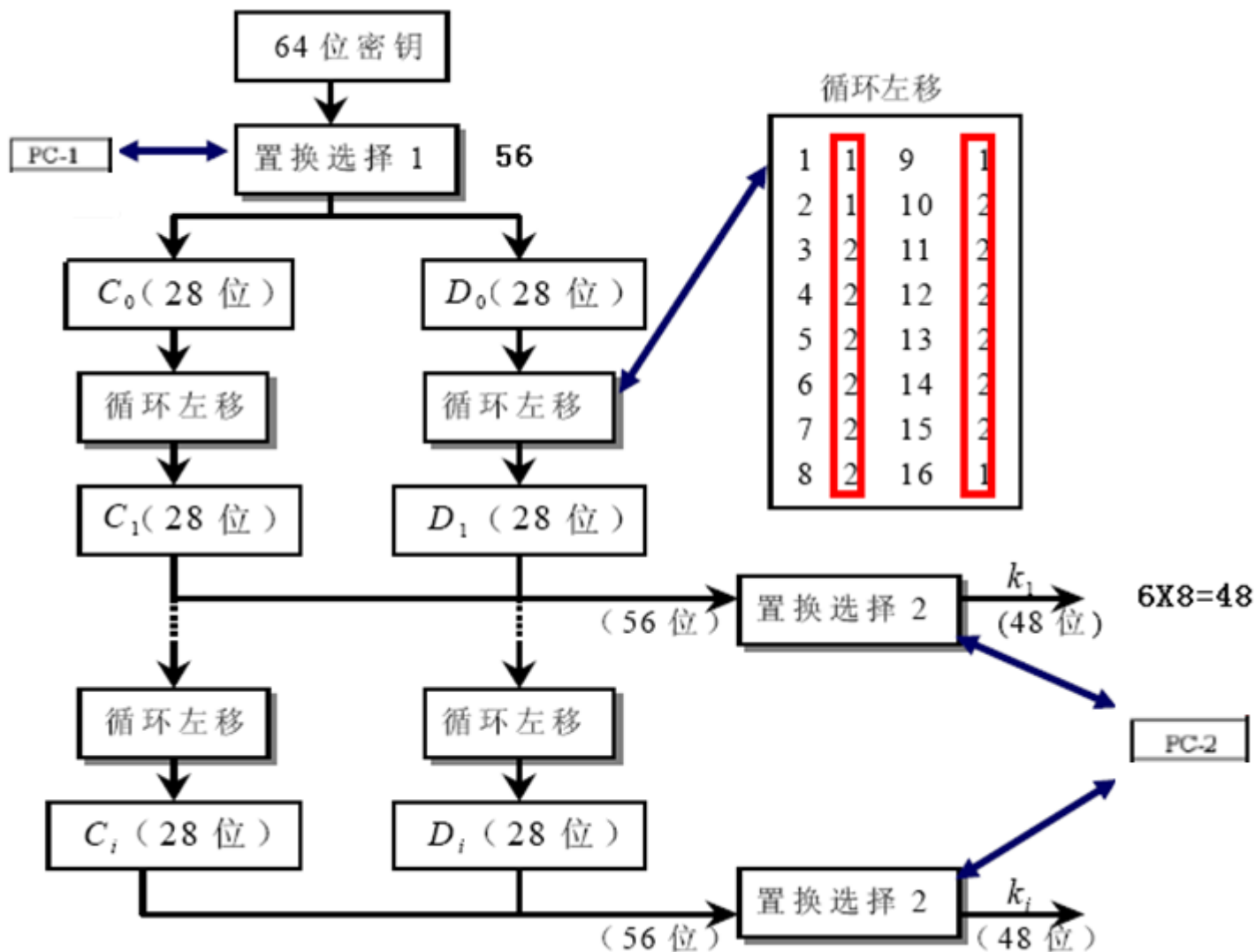
第 i 次迭代	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
循环左移次数	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

设计者要使

$$\sum_{i=1}^{16} L_i = 28$$

思考：为什么

子密钥的生成细节



DES解密

- 解密是加密的逆过程
- 对Feistel框架密码，采用相同算法，但是子密钥使用的次序正好相反：
 - IP变换抵消加密的最后一步 IP^{-1} ;
 - 第一轮使用密钥 K_{16} ;
 - 第二轮使用密钥 K_{15} ;
 -
 - 第十六轮使用密钥 K_1 ;
 - IP^{-1} 变换抵消加密的第一步IP;
 - 获得解密明文。

DES的解密

- DES的解密算法和加密算法完全相同
- 只是各子密钥的使用顺序相反，即为 $k_{16}, k_{15}, k_{14}, \dots, k_2, k_1$ 。算法也是循环右移产生每一圈的子密钥，每次右移动的位数为

Round number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Bits rotated	0	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

DES小结

- 用S-盒实现小块的非线性变换,达到混乱目的
- 用置换P实现大块的非线性变换,达到扩散目的
- DES的安全性完全依赖于所用的密钥,这是划时代的
- 密文与明文、密文与密钥的相关性:
 - 在DES的编码过程,可使每一密文比特都是所有明文比特和所有密钥比特的复杂混合函数,要达到这一点,DES至少要迭代5轮。
 - 人们也用 χ^2 -检验证明: DES迭代8轮以后,就可认为输出与输入不相关了

FEISTEL型分组码的设计考虑回顾



S-盒的
设计

P-置换
的设计

轮函数
的设计

DES安全性分析

弱密钥与半
弱密钥

密钥长度的
争论

F函数(尤其
是S-Box)设
计原理未知

DES的破译

雪崩效应--AVALANCHE EFFECT

- 明文或密钥的1比特的变化，引起密文许多比特的改变
- 加密算法的关键性能之一
- 希望明文或密钥的1比特变化，会使半数密文比特发生变化；否则，可能存在方法减小待搜索的明文和密钥空间
- DES密码有良好的雪崩效应

DES特性分析-1

- 明文或密钥的一点小的变动应该使密文发生一个大的变化。

- 给定明文改变1bit

00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000

10000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000

- 给定密钥不变

0000001 1001011 0100100 1100010 0011100
0011100 0011100 0110010

Change in Plaintext	
Round	Number of bits that differ
0	1
1	6
2	21
3	35
4	39
5	34
6	32
7	31
8	29
9	42
10	44
11	32
12	30
13	30
14	26
15	29
16	34

明文变化1BIT



DES特性分析-2

■ 给定明文不变

```
01101000 10000101 0010111 01111010 00010011  
01110110 11101011 10100100
```

■ 给定密钥变化1bit

```
1110010 1111011 1101111 0011000 0011101  
0000100 0110001 1101110  
0110010 1111011 1101111 0011000 0011101  
0000100 0110001 1101110
```

Change in Key	
Round	Number of bits that differ
0	0
1	2
2	14
3	28
4	32
5	30
6	32
7	35
8	34
9	40
10	38
11	31
12	33
13	28
14	26
15	34
16	35

密钥变化1BIT



DES的弱密钥

原因：初始密钥被分成两部分，每部分都单独移位



如果每一部分的每一位都是0或都是1，则每一轮的子密钥都相同



DES存在4个弱密钥

DES的弱密钥

- ◆ 至少有4个“弱密钥”： $E_k(E_k(m))=m$
 - 0101010101010101, 1F1F1F1F0E0E0E0E
 - 0E0E0E0E0F1F1F1F, FEFEFEFEFEFEFEFEF
- ◆ 至少有6对“半弱密钥”： $E_k(E_{k'}(m))=m$
 - 01FE01FE01FE01FE \leftrightarrow FE01FE01FE01FE01
 - 1FE01FE00EF10EF1 \leftrightarrow E01FE01FF10EF10E
 - 01E001E001F101F1 \leftrightarrow E001E001F101F101
 - 1FFE1FFE0EFE0EFE \leftrightarrow FE1FFE1FFE0EFE0E
 - 011F011F010E010E \leftrightarrow 1F011F010E010E01
 - E0FEE0FEEF1FEF1FE \leftrightarrow FEEOFEEOFEF1FEF1
- ◆ 这里每字节的最低位用作奇偶校验位

作业（课程项目两个）

- ◆ 仿照DES的思路设计一种密码算法
- ◆ 编写软件分析DES的雪崩效应，即：当密钥固定时，研究密文的某一个bit，明文的哪些bit变化，将导致密文该bit为0，或者为1
- ◆ 一周后提交

DES的分析原理

■ 利用加密算法的深层结构

- 搜集加密信息
- 最终设法恢复部分或全部子密钥的位
- 如果必要的话对其余部分再辅以穷举搜索

■ 本质上是统计分析，包括

- 差分分析、线性分析、相关密钥攻击

■ 结论：DES不能抵御差分分析、线性分析

DES的分析方法

- ① 暴力破解
- ② 分布式计算
- ③ 专用设备破解(破解机)
- ④ 差分密码分析法
- ⑤ 线性密码分析法
- ⑥ 时间与数据量折衷法

差分密码分析 DIFFERENTIAL CRYPTANALYSIS

■ 历史

- 1990年，Murphy、Biham和Shamir首次提出，是第一种可以以少于 2^{55} 的复杂性对DES进行破译的方法
- 需要 2^{47} 个选择明文及对应密文
- 1974年IBM的DES研究团队就发现了差分攻击，并在S盒子和置换P的设计中加以考虑

■ 原理

- 分析明文对的差异和密文对的差异之间的关系
- 确定轮运算的子密钥，从而恢复某些密钥比特

差分密码分析

- 是一种**选择明文**攻击，考查那些有特定差分的明文对及其对应的密文对，分析明文差分在通过轮扩散时的演变。
- 首先，随机选取**符合特定差分条件**的一对明文。然后，根据输出密文的差分，按照不同的概率分配给不同的密钥。随着分析的密文对越来越多，其中最可能的一个密钥将显现出来，这就是正确的密钥。
- 差分分析针对DES和其他类似有固定S-盒的算法，极大地依赖于S-盒的结构。

线性密码分析 LINEAR CRYPTANALYSIS

- 是一种已知明文攻击，使用线性近似值来描述分组密码的操作。
- 基本原理是寻找明文、密文和密钥间的有效线性逼近式，当该逼近式的线性偏差足够大时，就可以由一定量的明密文对推测出部分密钥信息。线性分析的关键是确定有效线性逼近式的线性偏差和线性组合系数。
- 理论上讲，应当利用明文、密文和原始密钥间的关系，但实际上寻找这种有效线性逼近式是困难的。通常的做法是寻找明文、密文和子密钥间的有效线性逼近式，并假设子密钥是独立的。

线性密码分析

- Matsui在1993年提出
- 攻击16轮DES需 2^{43} 个已知明文
- 基本原理：寻找密码算法的有效线性近似表达式
 - 令明文分组为 $P[1], \dots, P[n]$, 密文分组为 $C[1], \dots, C[n]$, 密钥为 $K[1], \dots, K[m]$
 - 线性密码分析的目标是找到如下有效线性方程：
 - $P[\alpha_1, \alpha_2, \dots, \alpha_a] \oplus C[\beta_1, \beta_2, \dots, \beta_b] = K[\gamma_1, \gamma_2, \dots, \gamma_c]$
 - 其中： $1 \leq a, b \leq n, 1 \leq c \leq m$, α, β 和 γ 表示比特位置
 - $A[i, j, \dots, k] = A[i] \quad A[j] \quad \dots \quad A[k]$
 - 方程成立的概率 p 离0.5越远，方程越有效。使 $|p-0.5|$ 最大的线性表达式称为最佳逼近式，相应的 p 称为最佳概率

强力攻击

- 所有密码算法最基本的攻击方法
- 依次尝试所有可能的密钥。密钥长度决定了可能的密钥数量，也决定了这种方法的可行性
- 对于DES，在成为标准之前就有一些关于其密钥长度的适当性的争论。正是它的密钥长度，而不是理论密码分析迫使它被后续算法所替代
- 在设计时，在与包括NSA在内的外部顾问讨论后，密钥长度被从128位减少到了56位，以适应在单芯片上实现算法

强力攻击的代价 — 上世纪

- ◆ 学术上曾有数个DES破解器（这些早期的设计并没有实现，至少没有公开的实现）：
 - 1977年迪菲和海尔曼提出造价约2千万美元的破解器，可以在一天内找到一个DES密钥
 - 1993年麦可·维纳设计造价约1百万美元的破解器，约在7小时内可找到一个密钥。
- ◆ 1997年，RSA赞助了一系列竞赛，奖励第一个成功破解以DES加密的信息的团队1万美元。Rocke Verser（<http://www.cs.cmu.edu/~dkindred/des/rocke-alg.html>），Matt Curtin和Justin Dolske领导的DESCHALL计划获胜（<http://www.interhack.net/projects/deschall/>），使用了数千台联接因特网的计算机的闲置计算能力
- ◆ 1998年，电子前言基金会（EFF，一个信息人权组织）制造了一台DES破解器，造价约\$250,000。可用稍多于2天的时间暴力破解一个密钥，显示了迅速破解DES的可能性

强力攻击的进展

- ◆ 确认的DES破解器是2006年由德国的鲁尔大学与基尔大学的工作组建造的COPACOBANA。一台COPACOBANA的造价大约是\$10,000
- ◆ 120片并列的XILINX Spartan3-1000型FPGA分为20个DIMM模块，每个模块包括6个FPGA。使用可重配置的FPGA使得这种装置也可以用于其它密码的破解
- ◆ 2007年，COPACOBANA的两个项目参与者组建的SciEngines公司改进了COPACOBANA
- ◆ 2008年，他们的COPACOBANA RIVYERA将破解DES的时间减少到了1天以内，使用128片Spartan-3 5000型FPGA
- ◆ SciEngines的RIVYEAR保持着使用暴力一天内破解法破解DES的纪录（2009年，<http://www.sciengines.com/company/news-a-events/74-des-in-1-day.html>）

日期	年份	事件
5月15日	1973	NBS第一次征集加密算法标准
8月27日	1974	NBS第二次征集加密算法标准
3月17日	1975	DES在“联邦公报”上发布并征集意见
8月	1976	DES的第一次研讨会
9月	1976	第二次研讨会，讨论DES的数学基础
11月	1976	DES被确认为标准
1月15日	1977	DES被作为FIPS标准FIPS PUB 46发布
	1983	DES第一次延长标准期限
	1986	HBO开始使用一个基于DES的电视卫星加密系统，Videocipher II
1月22日	1988	DES第二次延长标准期限，称为FIPS 46-1，取代FIPS PUB 46
7月	1990	毕汉姆和萨莫尔重新发现了微分密码分析，并将之应用到了一个15位的类DES密码系统
	1992	毕汉姆和萨莫尔发布了第一个复杂性小于暴力破解的理论攻击方法：微分密码分析。然而，这种方法仍然需要不现实的 2^{47} 选择明文。
12月30日	1993	DES作为FIPS 46-2第三次延长标准期限 ^[21]
	1994	试验了第一个实验性的DES密码分析，线性密码分析 ^{[22][23]}
6月	1997	DESCHAL计划第一次公开破解了DES加密的信息
7月	1998	EFF的DES破解器(Deep Crack)在56小时内破解了DES密钥
1月	1999	Deep Crack和distributed.net合作在22小时15分钟内破解了一个DES密钥
10月25日	1999	DES作为FIPS46-3第四次延长标准期限，其中规定优先使用3DES，而普通DES只允许在遗留的系统中应用 ^[24]
11月26日	2001	AES作为FIPS 197发布
5月26日	2002	AES标准开始生效
7月26日	2004	“联邦公报”发布了FIPS 46-3以及一系列相关标准被驳回的信息 ^[25]
5月19日	2005	NIST拒绝了FIPS 46-3标准 ^[26]
4月	2006	德国鲁尔大学和基尔大学基于FPGA的价值\$10,000的并行计算机COPACOBANA在9天内破解了DES ^[27] 在一年内，软件改进将平均时间降低到了6.4天。
11月	2008	COPACOBANA的下一代，RIVYERA将平均破解时间降低到了一天

DES的安全性争议

■ 颇多

- 用56比特密钥加密64比特数据
- 设计标准列入机密

■ 民间研究显示DES安全性很强

- 广泛应用在金融、遗产等领域
- 虽然差分攻击和线性分析攻击在理论上有效，但实现起来计算量仍很大

■ 曾是应用最广泛的分组密码技术

- 被AES取而代之

采用的技术手段分析—高次叠代

连续使用
两个或以
上的基本
密码变换

- 两次代替→更难分析的代替
- 两次置换→更难分析的置换
- 代替再置换→强度更高的密码

分组密码的发展



原有的改进

设计新的算法

DES的复合改进

证明DES 不能成为群

(K. W. Campbell and M. J. Wiener Proof that DES is not a group In Advances in Cryptology——Crpto' 92. Springer-Verlag , New York,1993)

多重DES，尤其是三重DES(TRIPLE DES)在普
遍使用

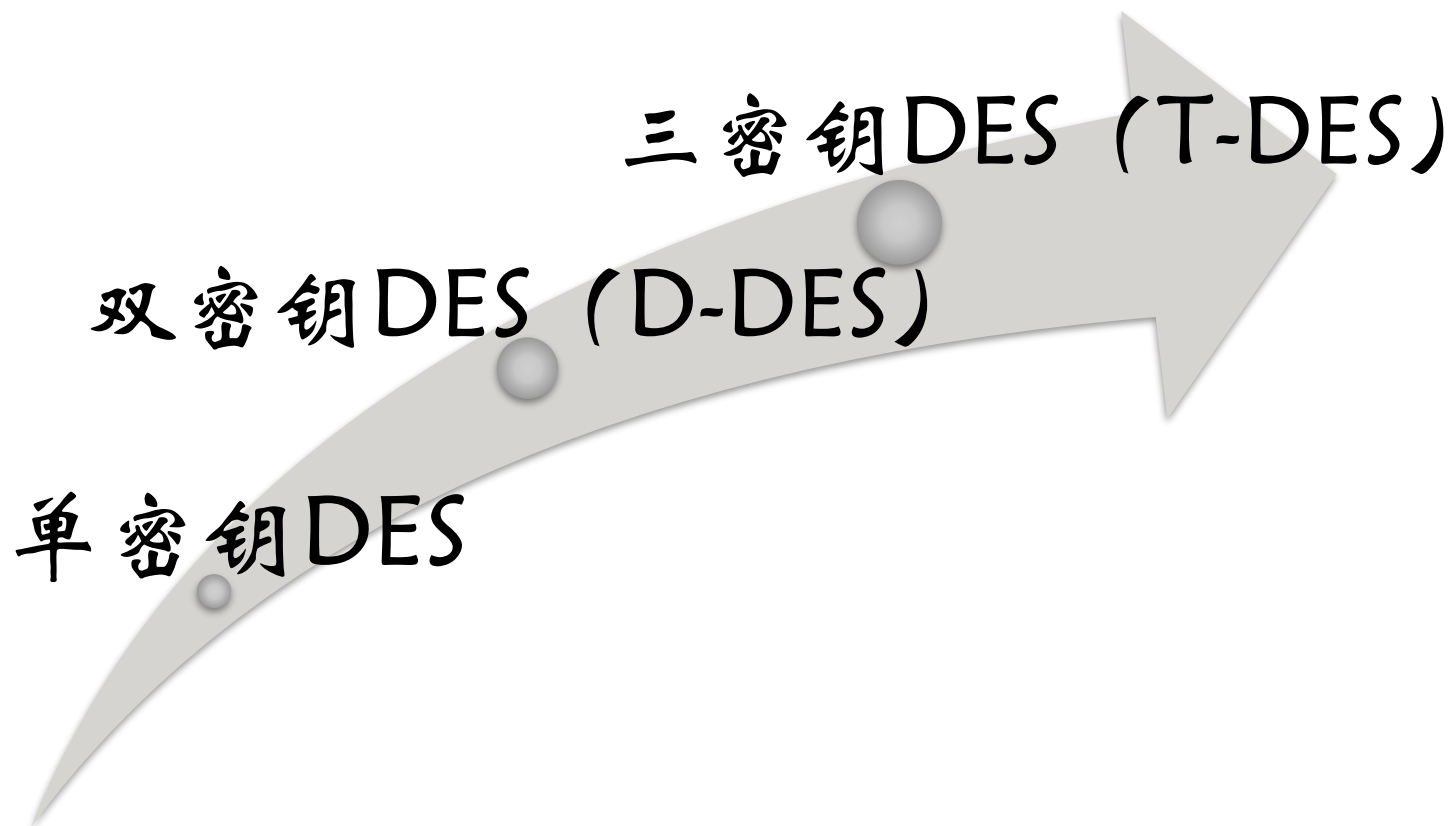
改进原理

某些运算 M 具有这样的性质，取自 M 的两个运算的合成仍属于 M ，也就是说这一运算形成群

若两个加密方法集合每一个都构成群，且可交换，则其的乘积加密也构成群

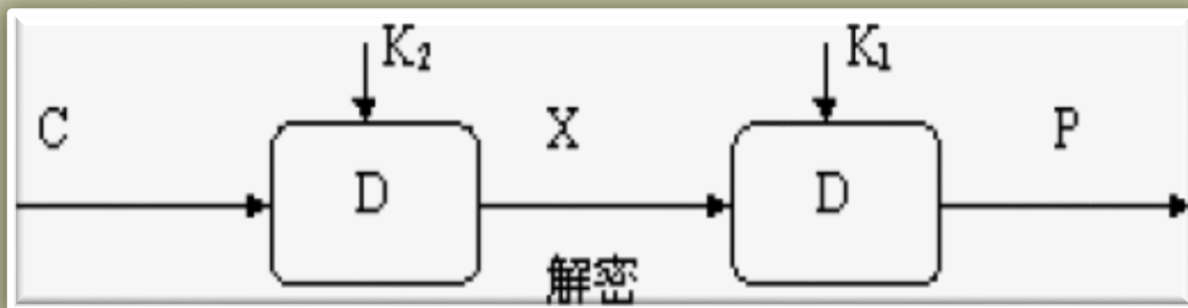
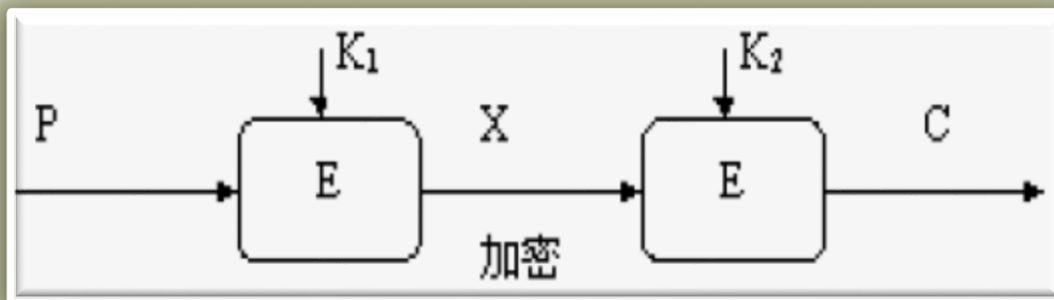
非群的情况，例如换位，执行一次“扩散”，多字母代替，执行一次“混乱”，因此可以被重复而且其组合复杂度进一步增加。等于是扩大了密钥空间

改进方法



双重DES (DOUBLE DES)

■ $C = E_{K_2}(E_{K_1}(P)) \Leftrightarrow P = D_{K_1}(D_{K_2}(C))$



TRIPLE-DES的四种模型

双密钥构造

- DES-EEE2: 两个不同密钥, $K1=K3$
- DES-EDE2: 两个不同密钥, $K1=K3$

三密钥构造

- DES-EEE3: 三个不同密钥, 顺序使用三次加密算法
- DES-EDE3: 三个不同密钥, 依次使用加密-解密-加密算法

双密钥的T-DES

由IBM设计,可
与常规加密算
法兼容

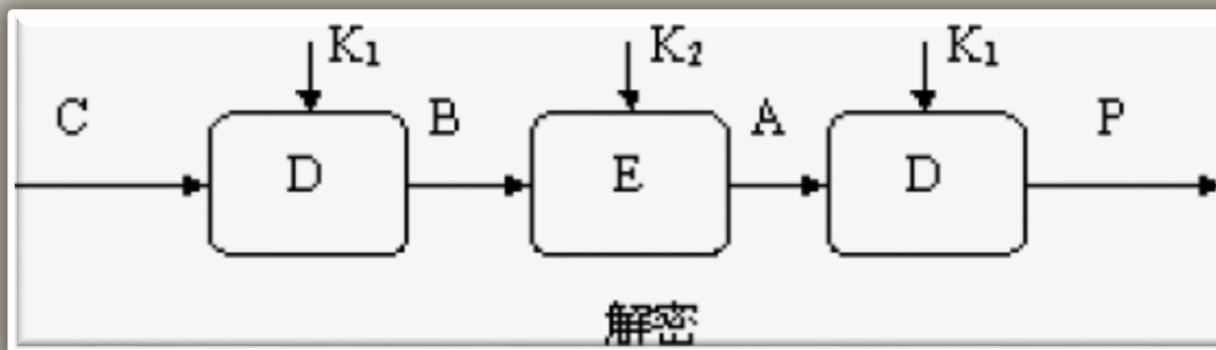
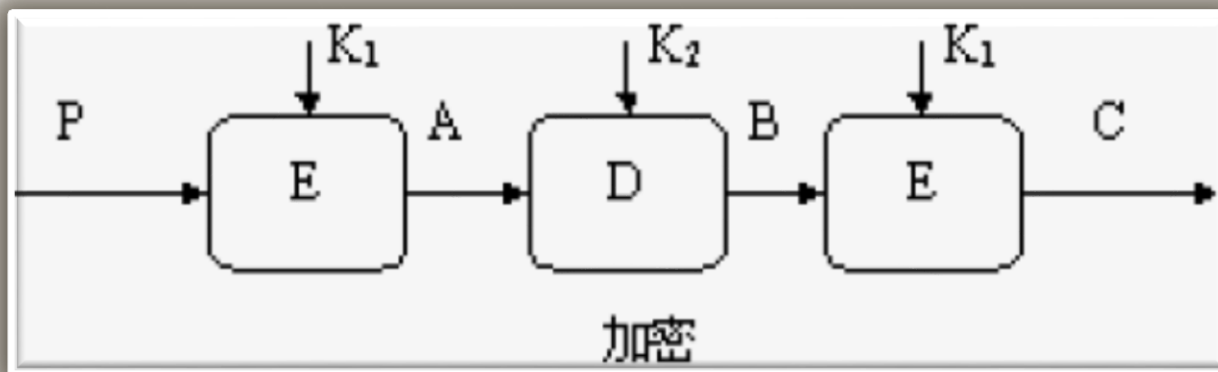


较流行,已被用
于密钥管理标
准(ANSX9.17和
ISO8732)



缺乏有效攻击
方法

举例：双密钥的T-DES



三密钥的T-DES

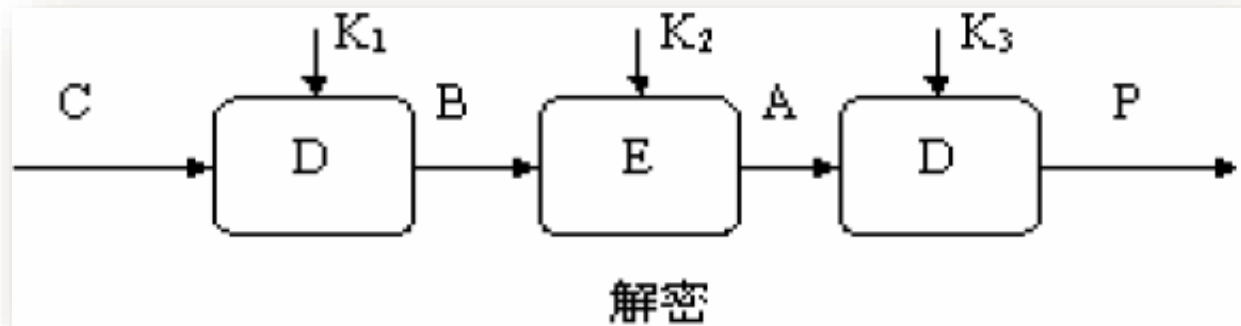
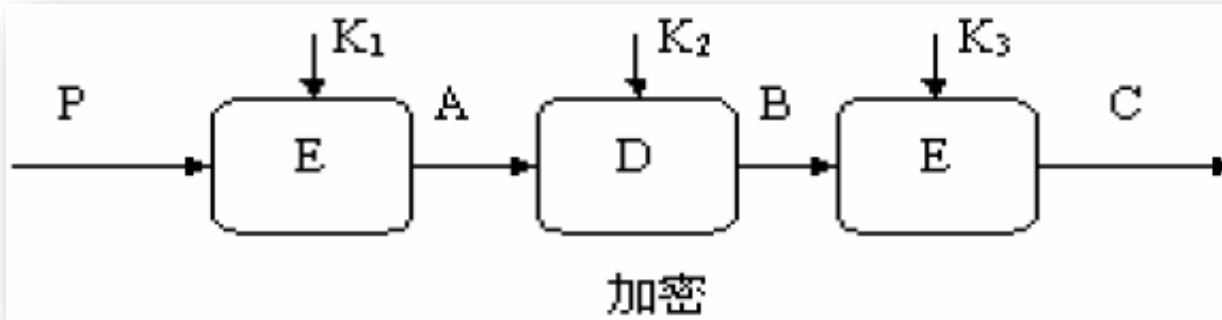
密钥有效长度：
168位

与DES的兼容性可
以通过令 $K3=K2$
或 $K1=K2$ 得到

许多基于Internet
的应用里用到：
PGP和S/MIME

三密钥T-DES举例

○ $C = E_{K_3}(D_{K_2}(E_{K_1}(P))) \Leftrightarrow P = D_{K_3}(E_{K_2}(D_{K_1}(C)))$



Comparison of Different Forms of DES Multiple Encryption

# of Encryptions	# of Keys	Computation	Storage	Type of Attack
single	1	2^{56}	-	known plaintext
single	1	2^{38}	2^{38}	chosen plaintext
single	1	-	2^{56}	chosen plaintext
double	2	2^{112}	-	known plaintext
double	2	2^{56}	2^{56}	known plaintext
double	2	-	2^{112}	chosen plaintext
triple	2	2^{112}	-	known plaintext
triple	2	2^{56}	2^{56}	2^{56} chosen plaintext
triple	2	2^{120-t}	2^t	2^t known plaintext
triple	2	-	2^{56}	chosen plaintext
triple	3	2^{112}	2^{56}	known plaintext
triple	3	2^{56}	2^{112}	chosen plaintext

NOTE : the most secure form of multiple encryption is triple-DES with three distinct keys

下次内容

- 作业讲评
- AES