PART 4

分组密码和数据加密标准

目 录 CONTENTS

4.1 传统分组密码结构

分组密码的原理

➢ 分组密码

- 是一种加/解密算法，将输入的明文分组当做一个整体处理，输出一个等长的密文分组。

Feistel

Feistel 密码结构是用于分组密码中的一种对称结构。以它的发明者Horst Feistel 为名，而Horst Feistel 本人是一位物理学家兼密码学家，在他为 IBM 工作的时候，为Feistel 密码结构的研究奠定了基础。

很多密码标准都采用了Feistel 结构，其中包括DES。

Feistel 的优点：由于它是对称的密码结构，所以对信息的加密和解密的过程就极为相似，甚至完全一样。这就使得在实施的过程中，对编码量和线路传输的要求就减少了几乎一半。

分组密码的原理

Feistel密码结构的设计动机

- n比特明文M(n) → n比特密文C(n)

  映射/代换的总数 $2^n! \approx n \times 2^n$

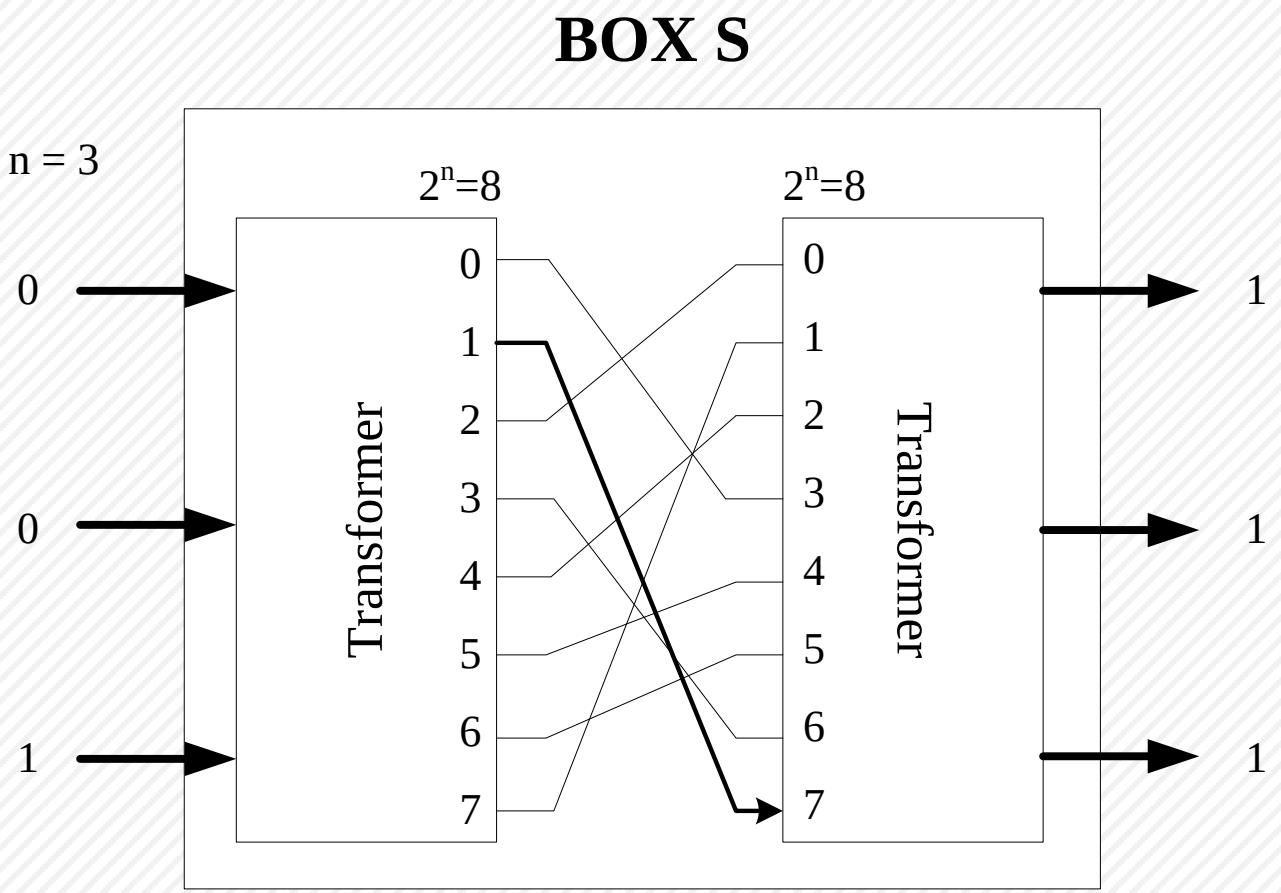
  具有上述代换个数的分组密码称为理想分组密码

- 映射/代换本身就是密钥，当n的规模不大时（如n＝4，密钥长度为64bit），密钥规模很大$n \times 2^n$，在实际中使用大规模分组的任意可逆代换密码是不可行的（理想分组密码在实际中难以实现）。

- Feistel提出在实际中所需的分析密码体制应该是对理想分组密码的一种近似体制。

- Feistel密码是对理想分组密码的近似。

- 图示：n＝3时，一个n位到n位的分组密码



Feistel密码

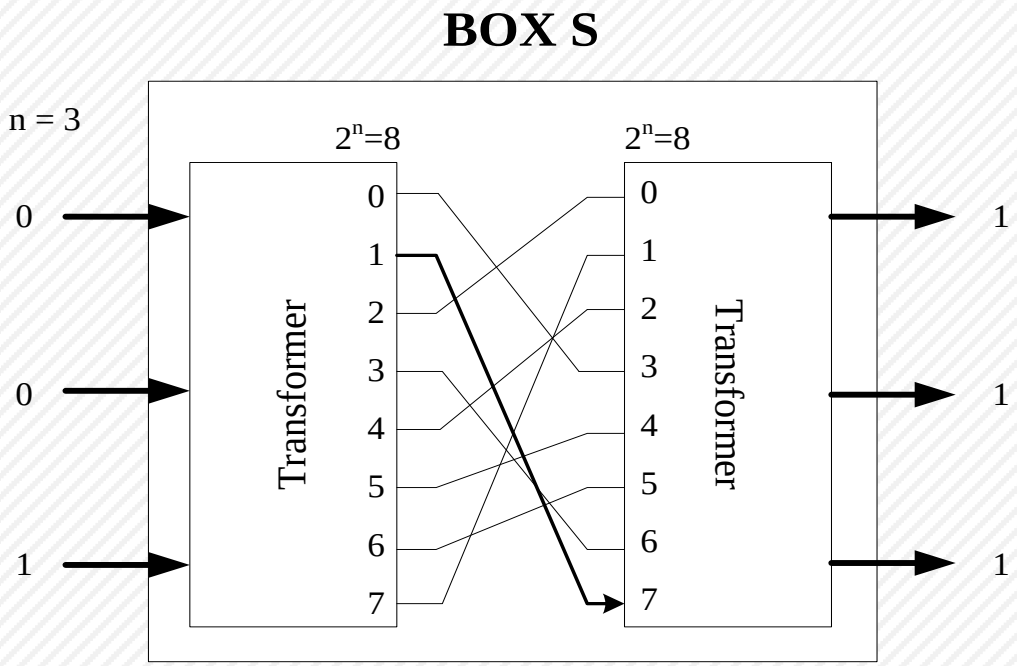
- 大多数传统分组加密算法都采用Feistel密码结构，包括DES在内。
- Feistel建议使用乘积密码来增强密码的强度。
- Feistel建议交替使用代换和置换，增强密码的扩散和混淆性能。

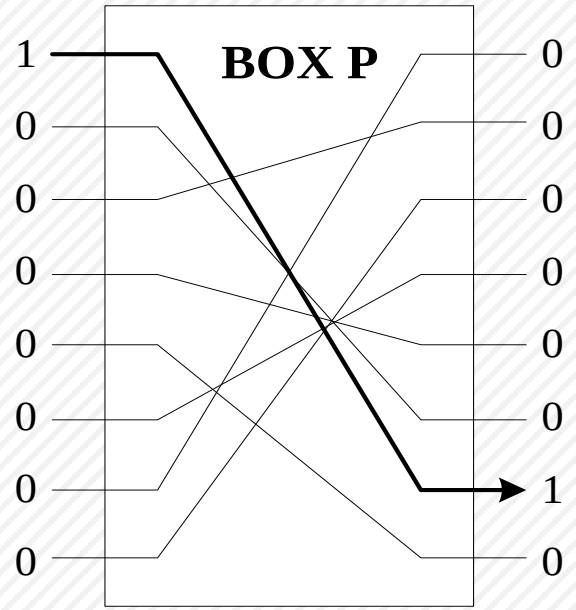
香农理论 1949

- 混淆Confusion
  - 尽可能使密文和加密密钥之间的关系变得复杂，以阻止攻击者发现密钥。

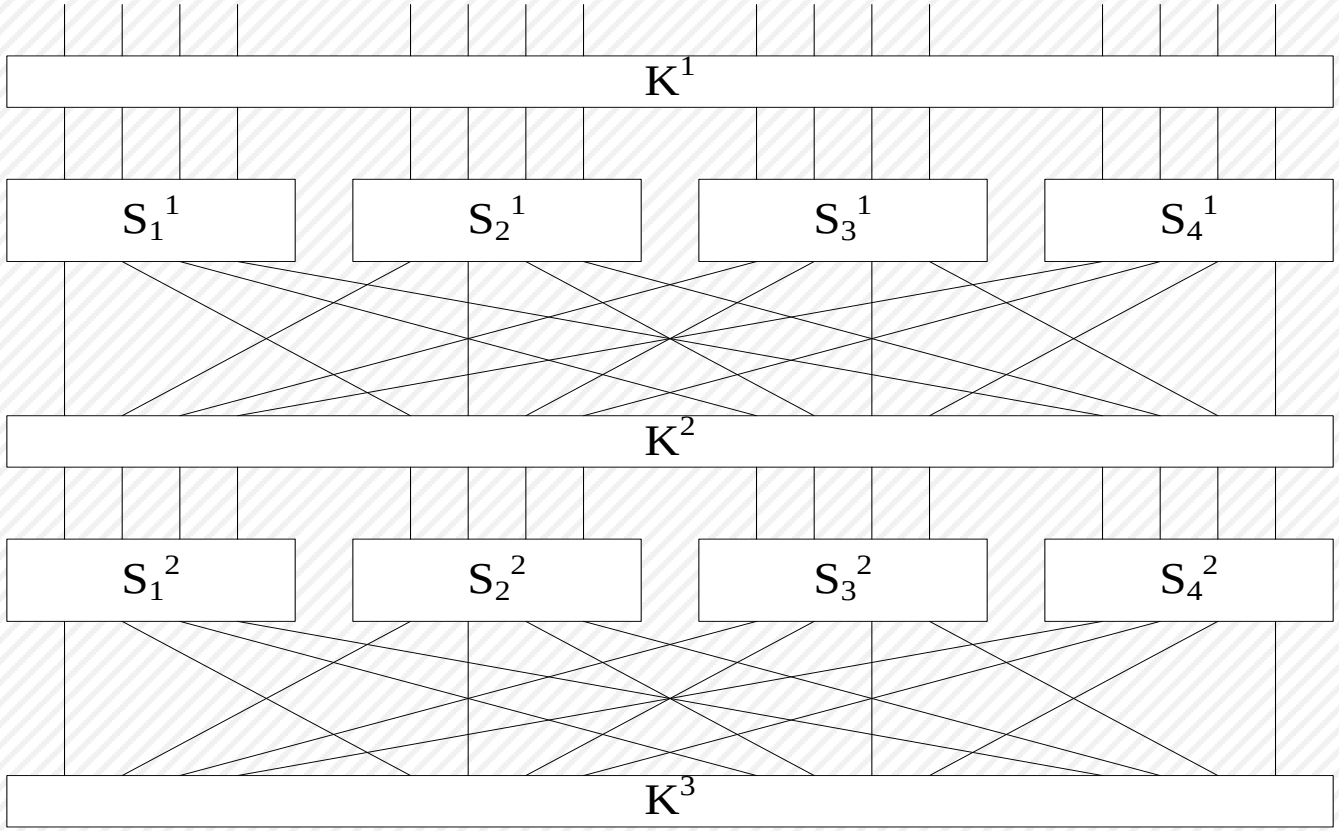
    密钥和密文关系为非线性关系
- 扩散Diffusion
  - 让每个明文数字尽可能地影响多个密文数字，使明文的统计特性消散在密文中。明文和密文间关系为非线性关系
- 乘积密码Product Cipher
  - 在单个加密机制中依次使用两个或两个以上不同类型的基本密码（如：代换和置换），所得结果的密码强度将强于每个单个密码的强度。

代换—置换网络


BOX S
n = 3
2^n=8
2^n=8
Transformer
Transformer
0
0
1
0 1 2 3 4 5 6 7
0 1 2 3 4 5 6 7
1
1
1


BOX P
1 0 0 0 0 0 0 0
0 0 0 0 0 1 0

代换—置换网络



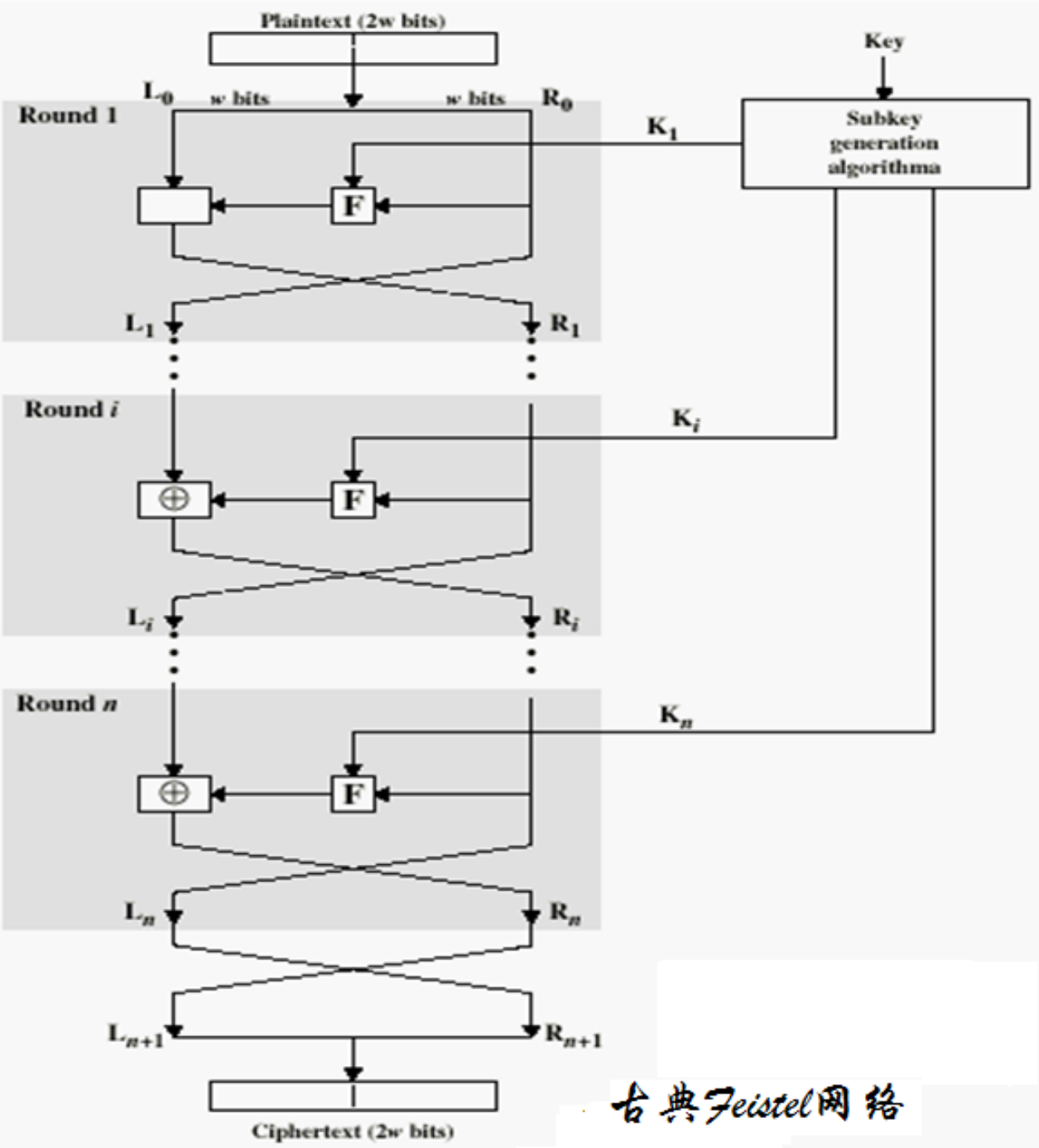
……

Feistel密码结构

- Feistel网络的实现依赖于以下参数的选择和特征：
- 分组长度
  - 分组越长意味着安全性越高（其他参数不变），但是会降低加/解密的速度。
  - 安全性的增加来自于更好的扩散性能。
- 密钥长度
  - 密钥越长意味着更高的安全性，但是会降低加/解密的速度。
  - 安全性的增加来自于更好的抗穷举攻击能力和更好的混淆性能。
- 迭代轮数
  - 单轮加密不能提供足够的安全性，而多轮加密可取得很高的安全性。

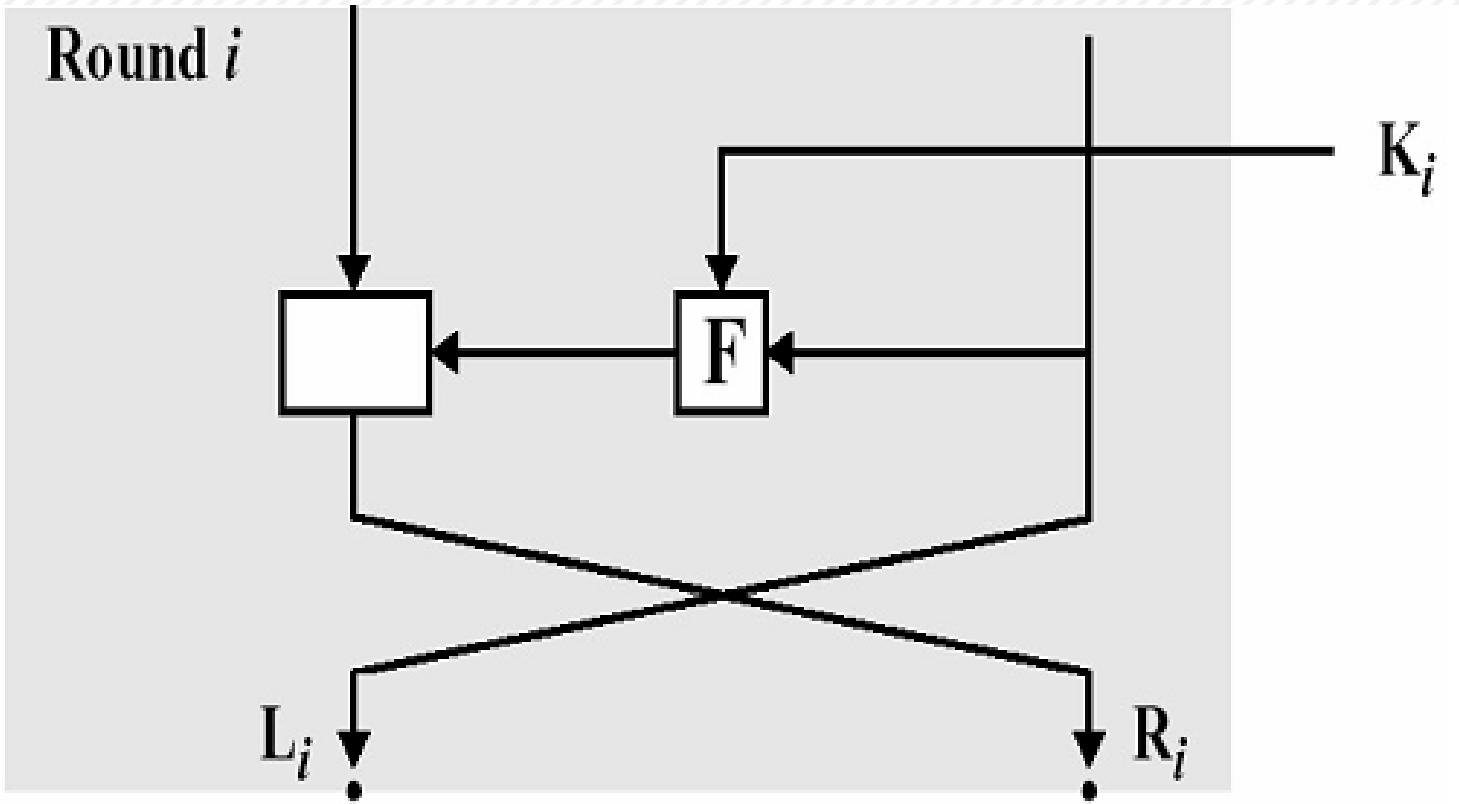
Feistel密码结构

- 子密钥生成算法
  - 子密钥生成算法越复杂，密码分析攻击就越困难。
- 轮函数
  - 轮函数越复杂，抗击密码分析的攻击能力就越强。
- 快速软件加/解密
  - 算法的执行速度是一个很受关注的指标。
- 简化分析难度
  - 尽管希望设计的算法能提高密码分析攻击的难度，但是算法本身的简洁明了有助于对其本身脆弱性的分析，从而设计出更强的算法。



古典Feistel网络

Feistel网络结构



Feistel密码结构

加密

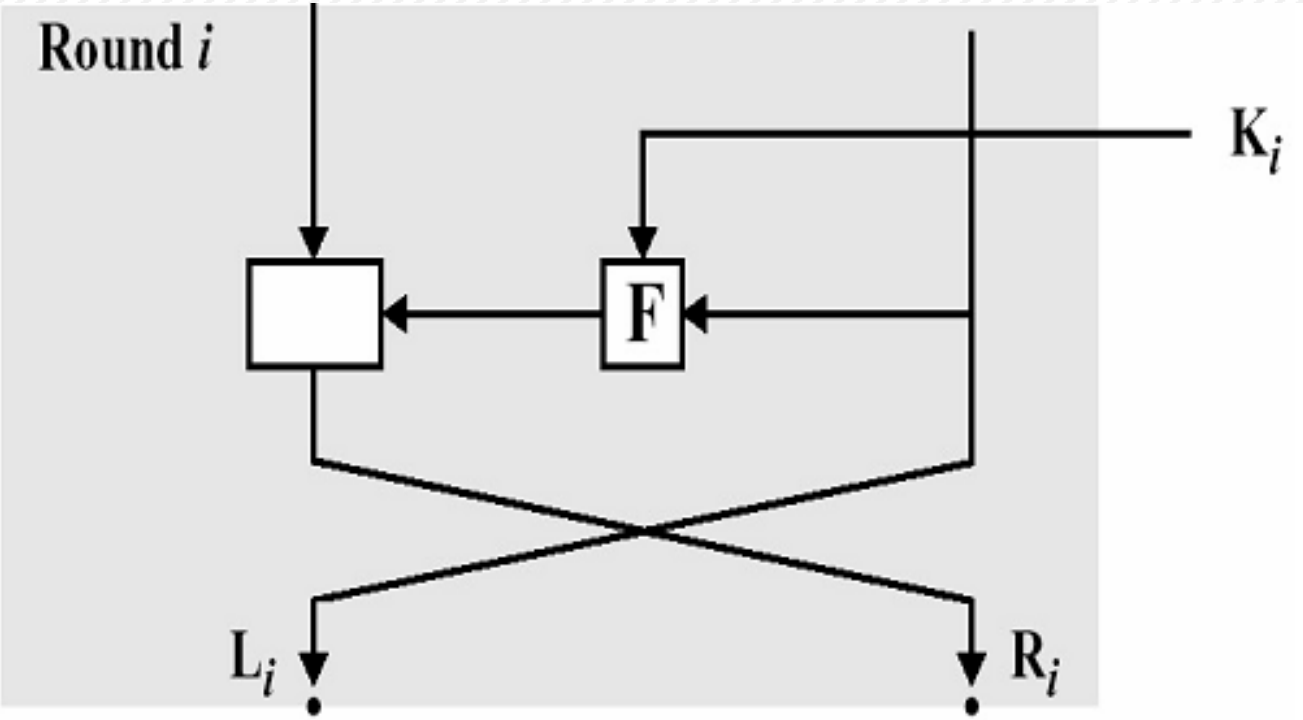
$$L^i = R^{i-1}$$

$$R^i = L^{i-1} \oplus f(R^{i-1}, K^i)$$

解密

$$L^{i-1} = R^i \oplus f(L^i, k^i)$$

$$R^{i-1} = L^i$$



4.2 数据加密标准

数据加密标准DES

DES的产生

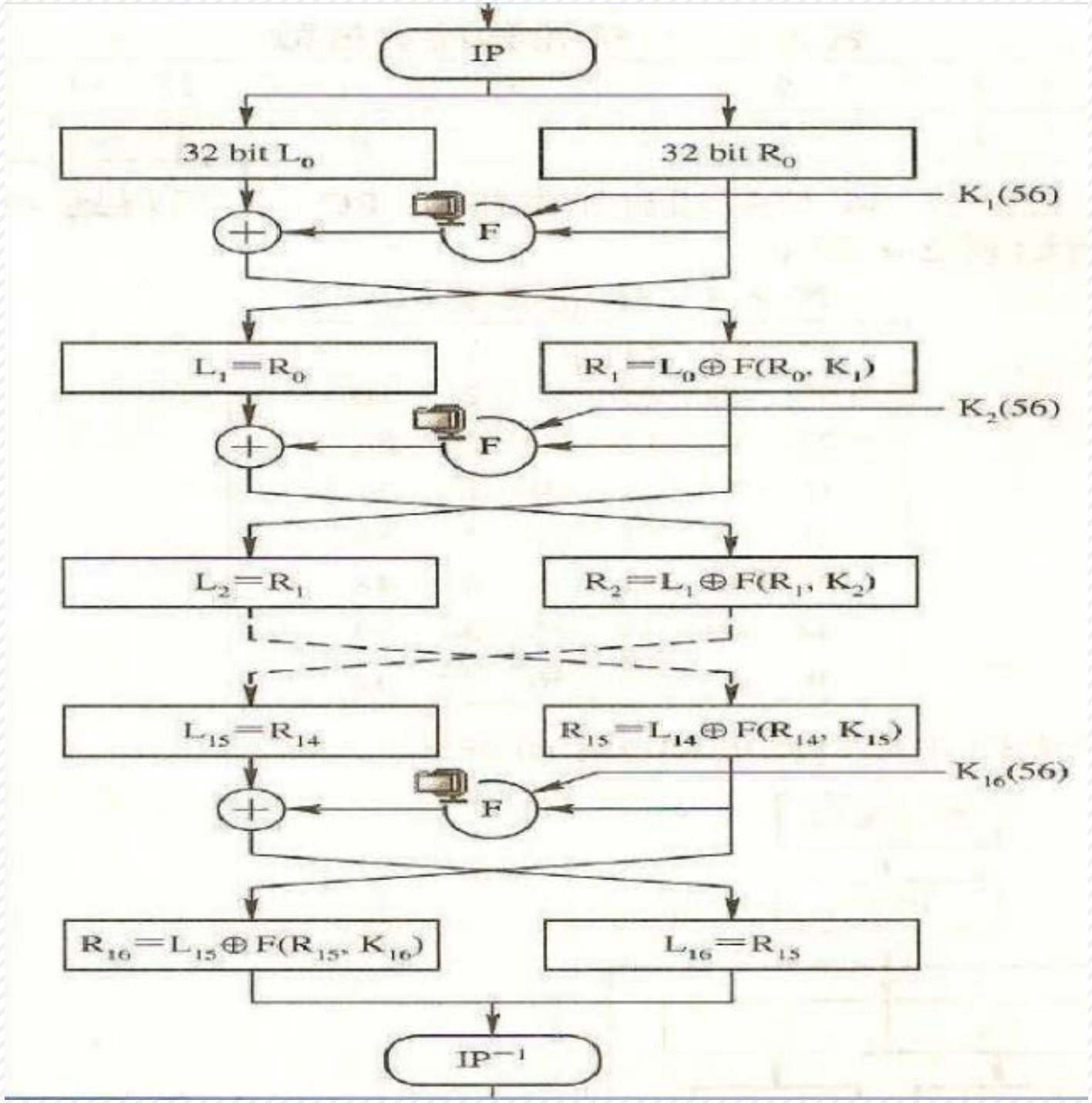
- 1973年5月15日,NBS（美国标准局）开始公开征集标准加密算法,并公布了它的设计要求:
  - (1)算法必须提供高度的安全性
  - (2)算法必须有详细的说明,并易于理解
  - (3)算法的安全性取决于密钥,不依赖于算法
  - (4)算法适用于所有用户
  - (5)算法适用于不同应用场合
  - (6)算法必须高效、经济
  - (7)算法必须能被证实有效
  - (8)算法必须是可出口的

- 1974年8月27日,NBS开始第二次征集,IBM提交了算法LUCIFER，该算法由IBM的工程师在1971~1972年研制。
- 1975年3月17日,NBS公开了全部细节
- 1976年,NBS指派了两个小组进行评价
- 1976年11月23日，采纳为联邦标准，批准用于非军事场合的各种政府机构
- 1977年1月15日,“数据加密标准”FIPS PUB 46发布

DES的应用

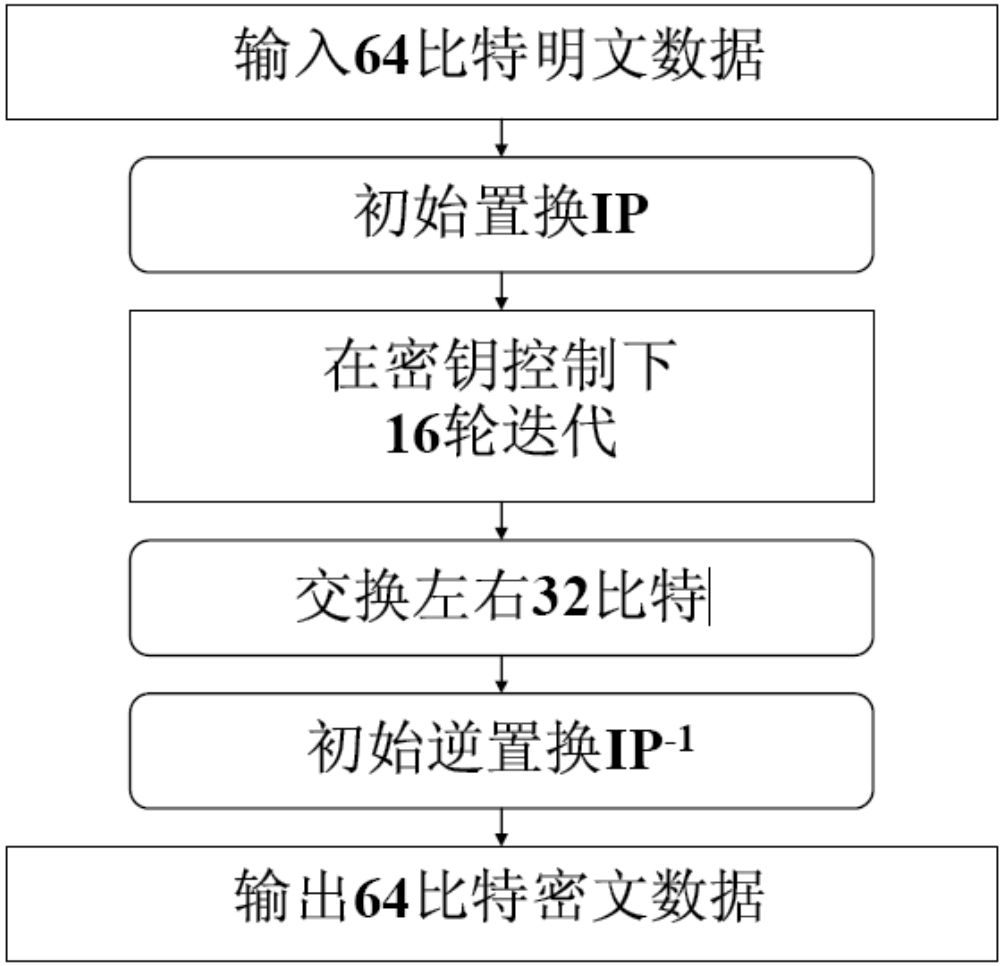
- 1979年，美国银行协会批准使用
- 1980年，美国国家标准局（ANSI）赞同DES作为私人使用的标准,称之为DEA（ANSI X.392）
- 1983年，国际化标准组织ISO赞同DES作为国际标准，称之为DEA-1
- 该标准规定每五年审查一次，计划十年后采用新标准
- 在1994年1月，决定1998年12月以后，DES将不再作为联邦加密标准。

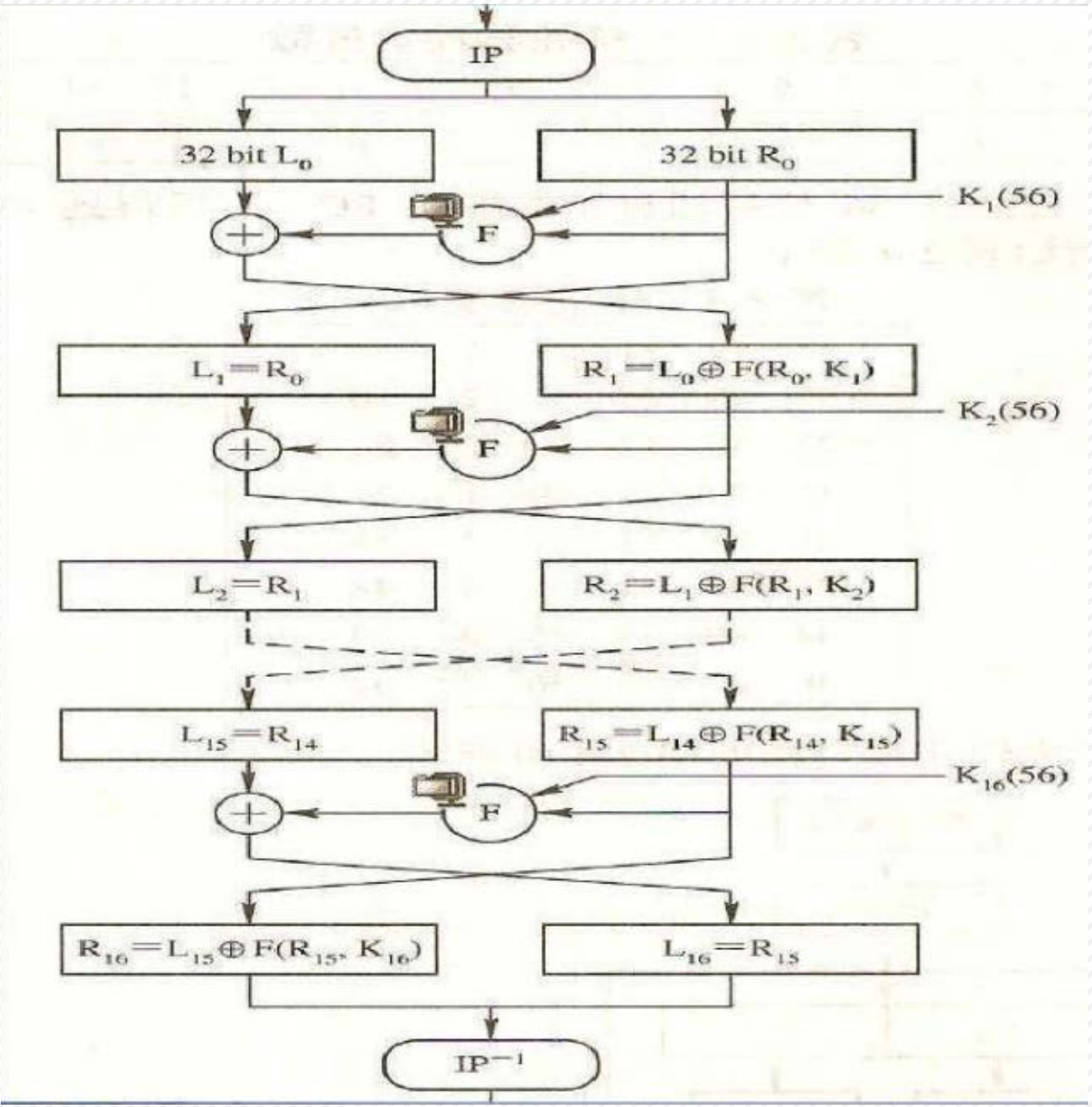
DES 示意图


IP
32 bit L_0
32 bit R_0
$K_1(56)$
F
$L_1 = R_0$
$R_1 = L_0 \oplus F(R_0, K_1)$
$K_2(56)$
F
$L_2 = R_1$
$R_2 = L_1 \oplus F(R_1, K_2)$
$L_{15} = R_{14}$
$R_{15} = L_{14} \oplus F(R_{14}, K_{15})$
$K_{16}(56)$
F
$R_{16} = L_{15} \oplus F(R_{15}, K_{16})$
$L_{16} = R_{15}$
IP^{-1}

DES描述

- DES利用56比特串长度的密钥K来加密长度为64位的明文，得到长度为64位的密文。





输入64比特明文数据

初始置换IP

在密钥控制下
16轮迭代

交换左右32比特

初始逆置换IP^{-1}

输出64比特密文数据

DES加解密过程

- 令*i*表示迭代次数，⊕表示逐位模2求和，*f*为加密函数。

DES的加密和解密过程表示如下。

加密过程：

$$L_0R_0 \leftarrow IP(<64bit\text{输入码}>)$$

$$L_i \leftarrow R_{i-1} \qquad i=1,2,\Lambda\ ,16$$

$$R_i \leftarrow L_{i-1} \oplus f(R_{i-1},k_i) \qquad i=1,2,\Lambda\ ,16$$

$$<64bit\text{密文}> \leftarrow IP^{-1}(R_{16}L_{16})$$

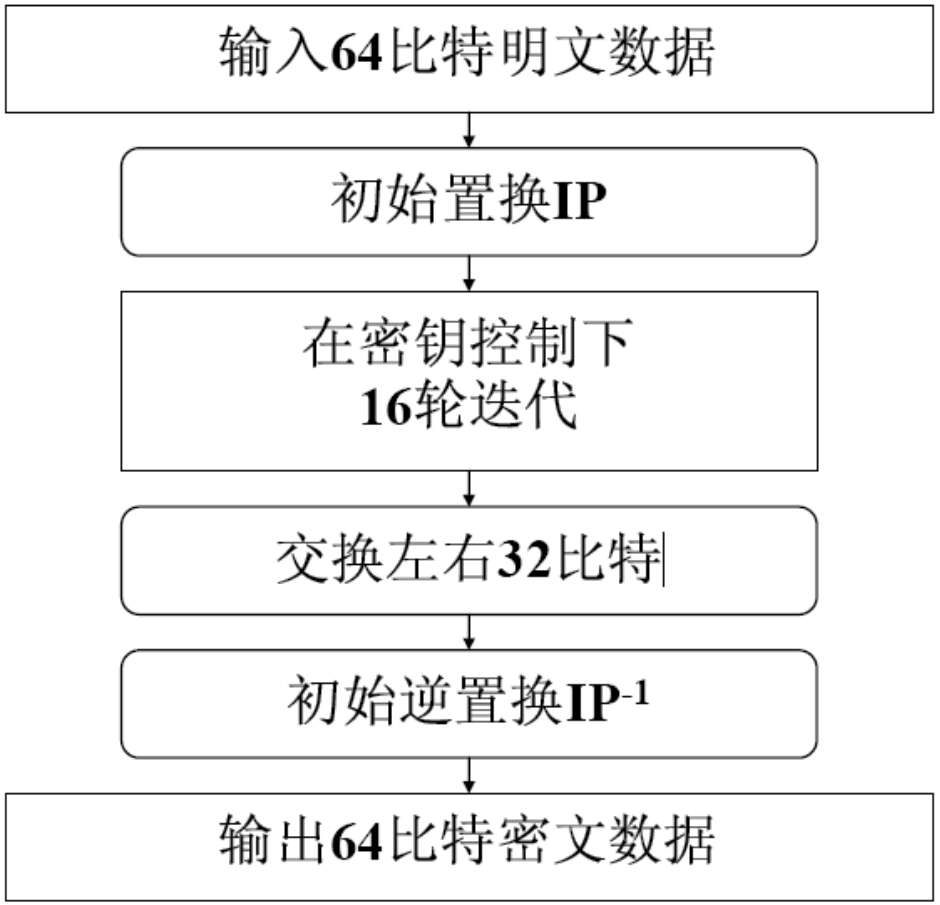
解密过程：

$$R_{16}L_{16} \leftarrow IP(<64bit\text{密文}>)$$

$$R_{i-1} \leftarrow L_i \qquad i=16,15,\Lambda\ ,1$$

$$L_i \leftarrow R_{i-1} \oplus f(R_{i-1},k_i) \qquad i=16,15,\Lambda\ ,1$$

$$<64bit\text{明文}> \leftarrow IP^{-1}(R_0L_0)$$



加密过程：

$$L_0R_0 \leftarrow IP(< 64bit\text{输入码} >)$$

$$L_i \leftarrow R_{i-1} \qquad i = 1,2,\Lambda\ ,16$$

$$R_i \leftarrow L_{i-1} \oplus f(R_{i-1}, k_i) \qquad i = 1,2,\Lambda\ ,16$$

$$< 64bit\text{密文} > \leftarrow IP^{-1}(R_{16}L_{16})$$

解密过程：

$$R_{16}L_{16} \leftarrow IP(< 64bit\text{密文} >)$$

$$R_{i-1} \leftarrow L_i \qquad i = 16,15,\Lambda\ ,1$$

$$L_i \leftarrow R_{i-1} \oplus f(R_{i-1}, k_i) \qquad i = 16,15,\Lambda\ ,1$$

$$< 64bit\text{明文} > \leftarrow IP^{-1}(R_0L_0)$$

初始置换IP和初始置换的逆置换IP-1

海南大学 Hainan university

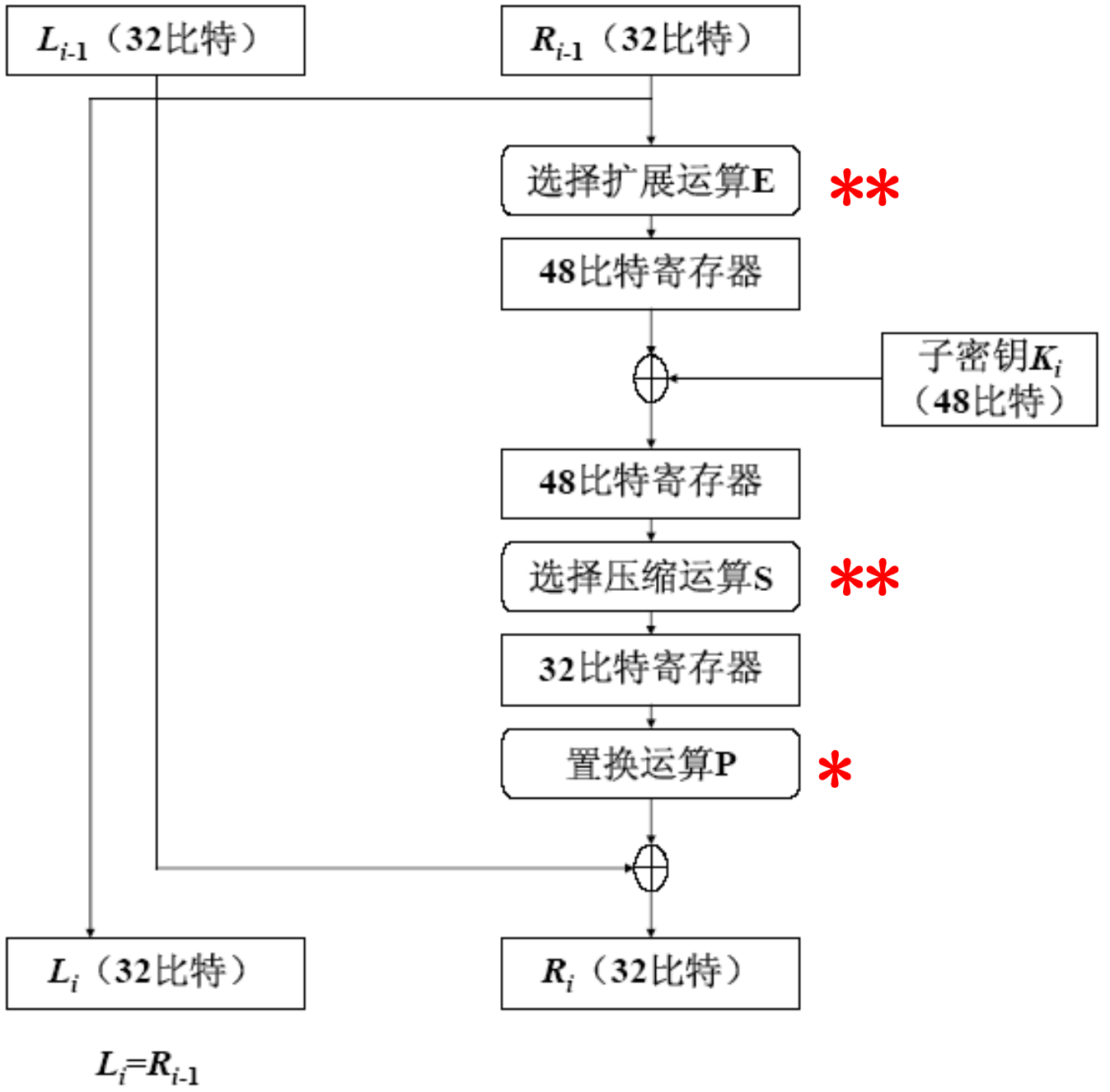
初始置换 IP

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

初始逆置换 IP^{-1}

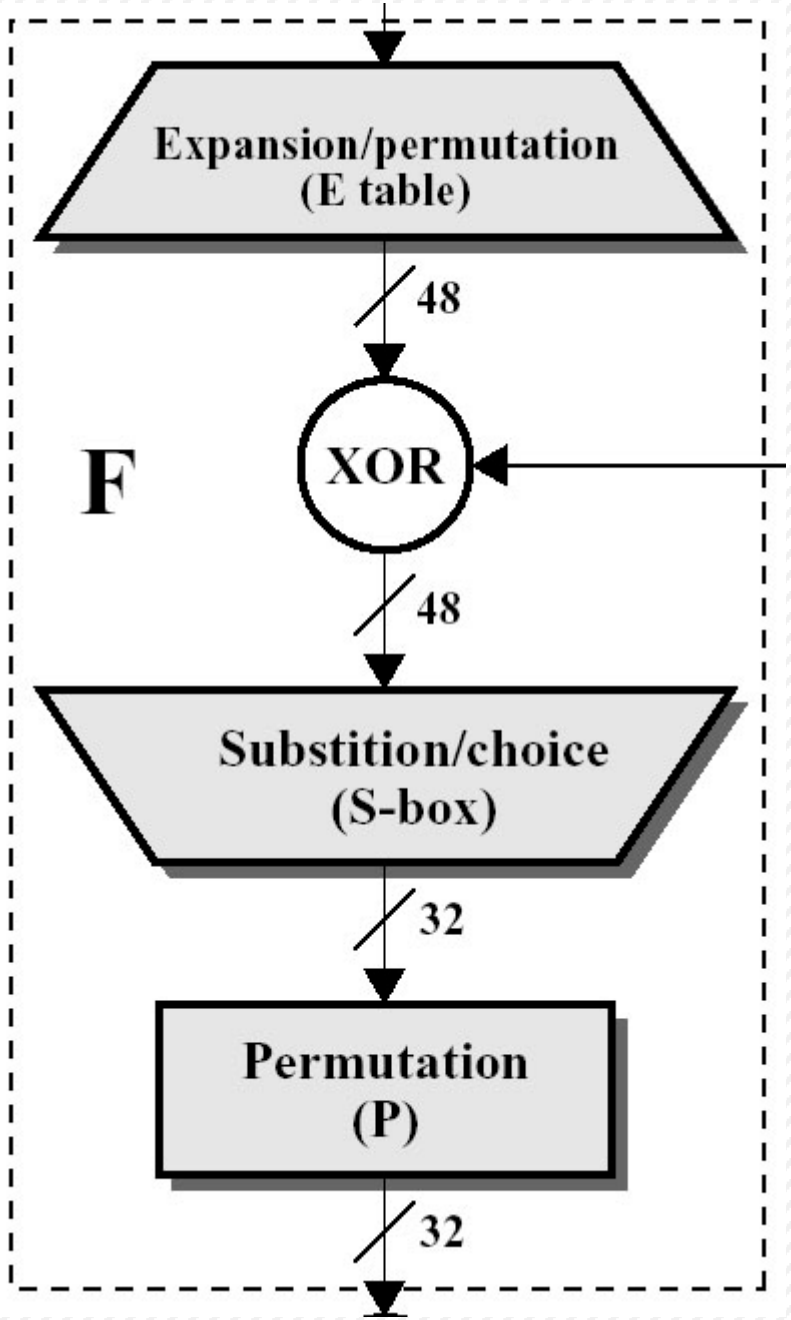
40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25

DES的一轮迭代


L_{i-1}（32比特）
R_{i-1}（32比特）
选择扩展运算E
**
48比特寄存器
子密钥K_i
（48比特）
48比特寄存器
选择压缩运算S
**
32比特寄存器
置换运算P
*
L_i（32比特）
R_i（32比特）
$L_i=R_{i-1}$

轮函数F

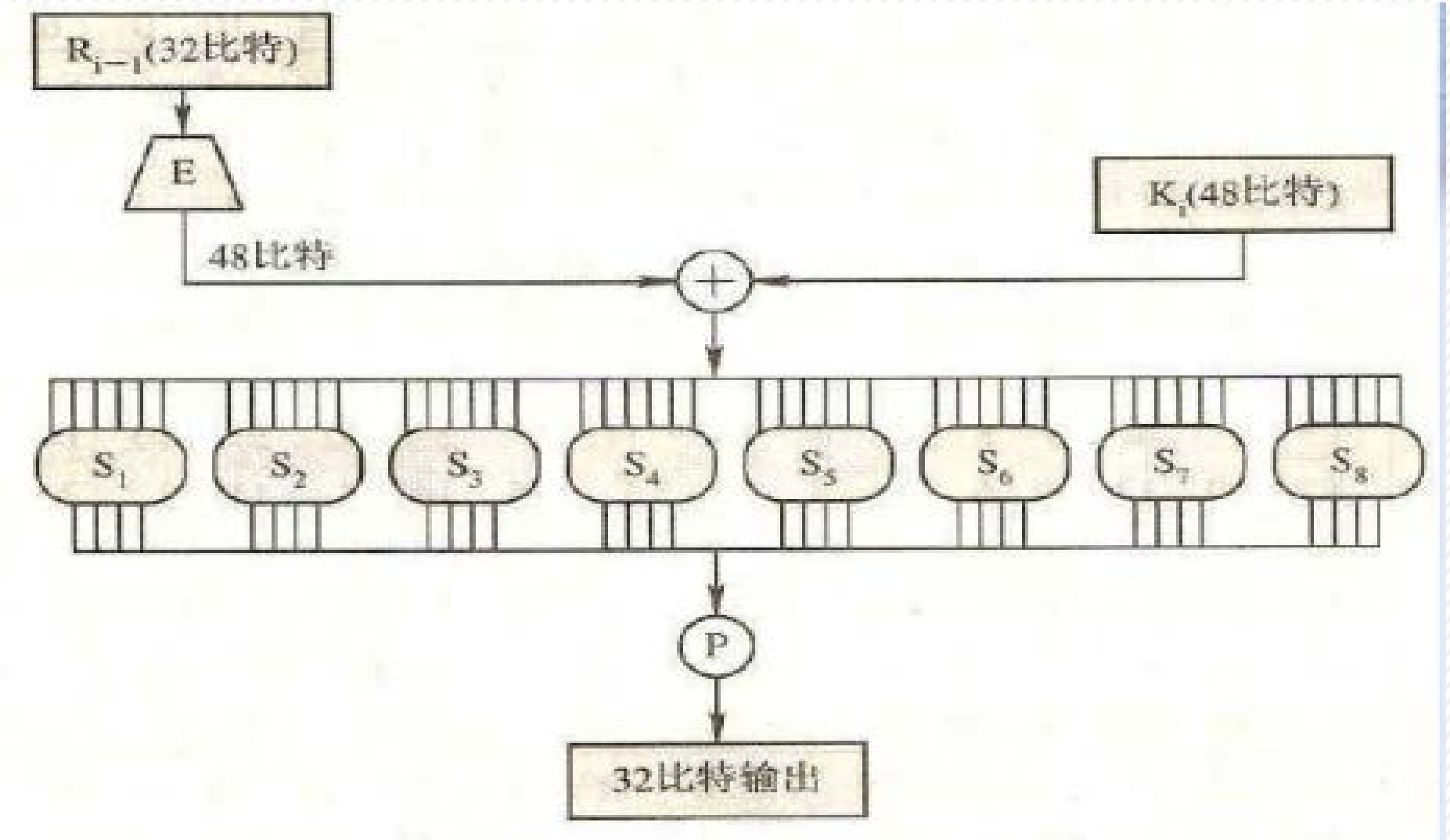
- 扩展: 32位→ 48位
- 8个S盒，每个S-box: 6位 → 4位
- 置换



扩展置换E

- 32 | 01 02 03 04 | 05
- 04 | 05 06 07 08 | 09
- 08 | 09 10 11 12 | 13
- 12 | 13 14 15 16 | 17
- 16 | 17 18 19 20 | 21
- 20 | 21 22 23 24 | 25
- 24 | 25 26 27 28 | 29
- 28 | 29 30 31 32 | 01

选择压缩/代换运算S



S盒

S盒是对阵密码系统分组加密算法中最核心的安全部件

- DES算法中
- 对每个盒，6比特输入中的第1和第6比特组成的二进制数确定的行，中间4位二进制数用来确定的列。
- 相应行、列位置的十进制数的4位二进制数表示作为输出。

例如:

输入为101001，则行数和列数的二进制表示分别是11和0100，即第3行和第4列。

第3行和第4列的十进制数为4，用4位二进制数表示为0100

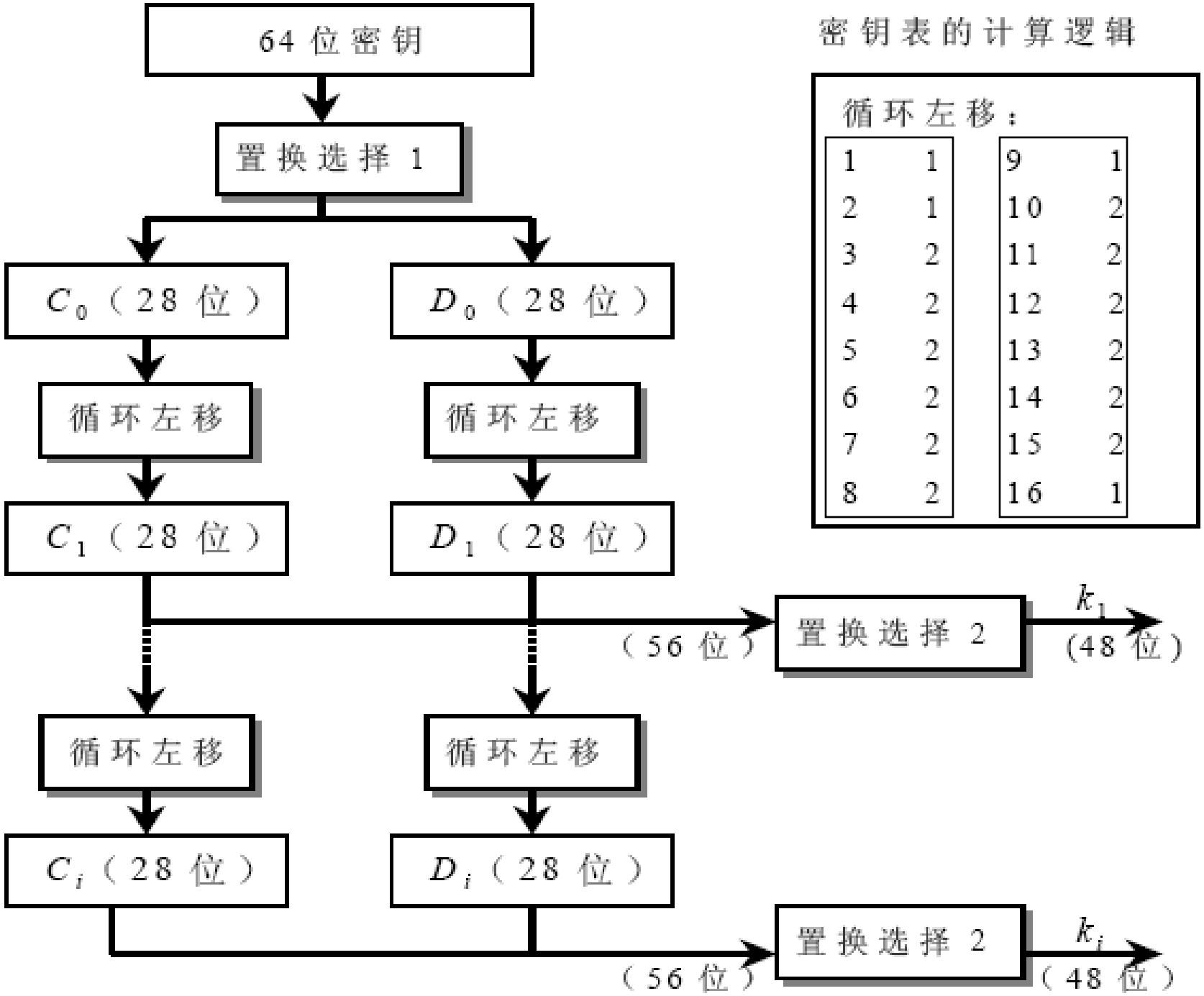
输出为0100。

	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

置换函数P-2

- 16 07 20 21 29 12 28 17
- 01 15 23 26 05 18 31 10
- 02 08 24 14 32 27 03 09
- 19 13 30 06 22 11 04 25

密钥的产生



密钥表的计算逻辑

循环左移：

1	1	9	1
2	1	10	2
3	2	11	2
4	2	12	2
5	2	13	2
6	2	14	2
7	2	15	2
8	2	16	1

置换选择1：舍弃 64位密钥中的奇偶校验位

密钥的产生

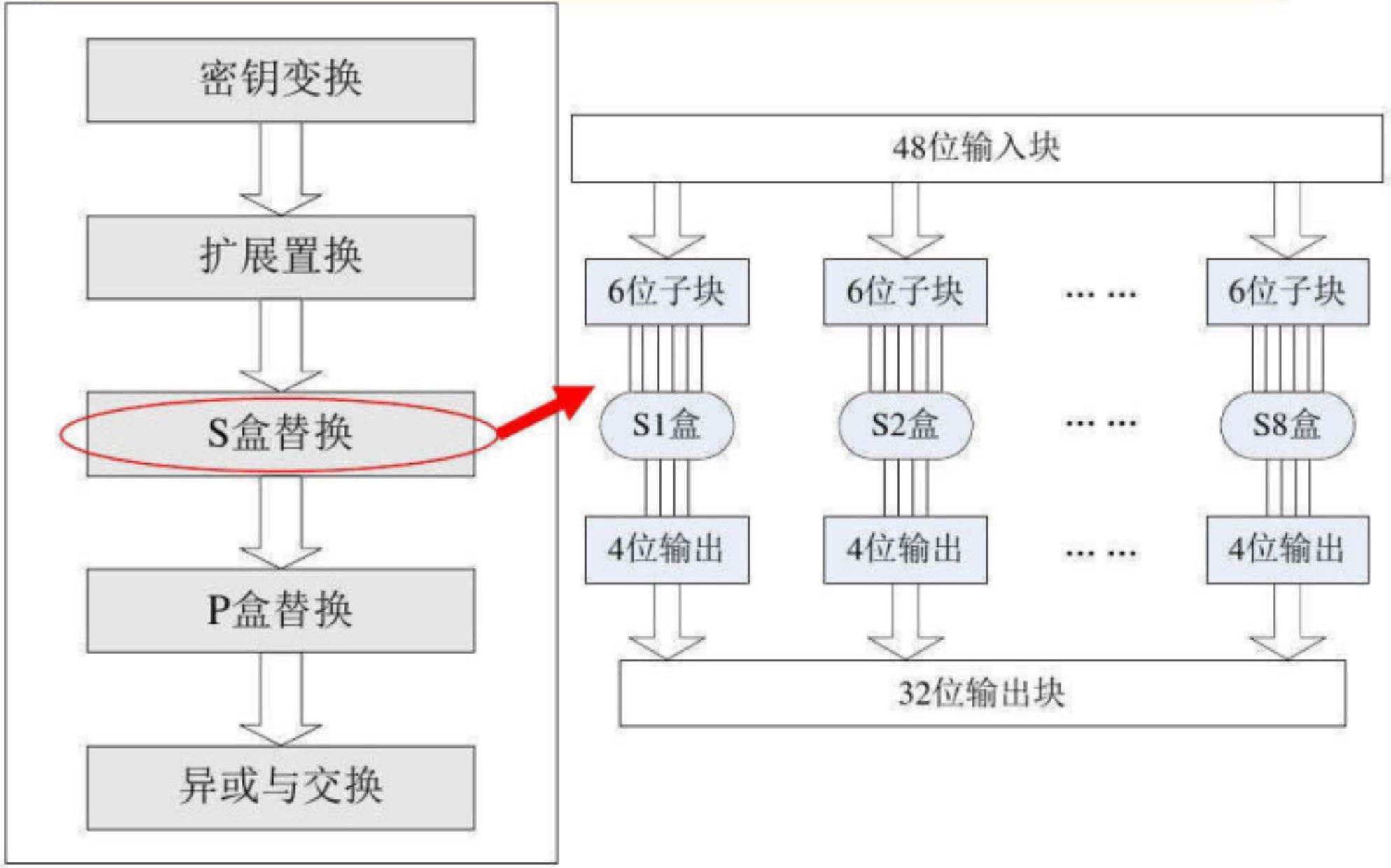
57	49	41	33	25	17	9
1	58	50	42	34	26	18
10	2	59	51	43	35	27
19	11	3	60	52	44	36
63	55	47	39	31	23	15
7	62	54	46	38	30	22
14	6	61	53	45	37	29
21	13	5	28	20	12	4

置换选择2：得到48位密钥

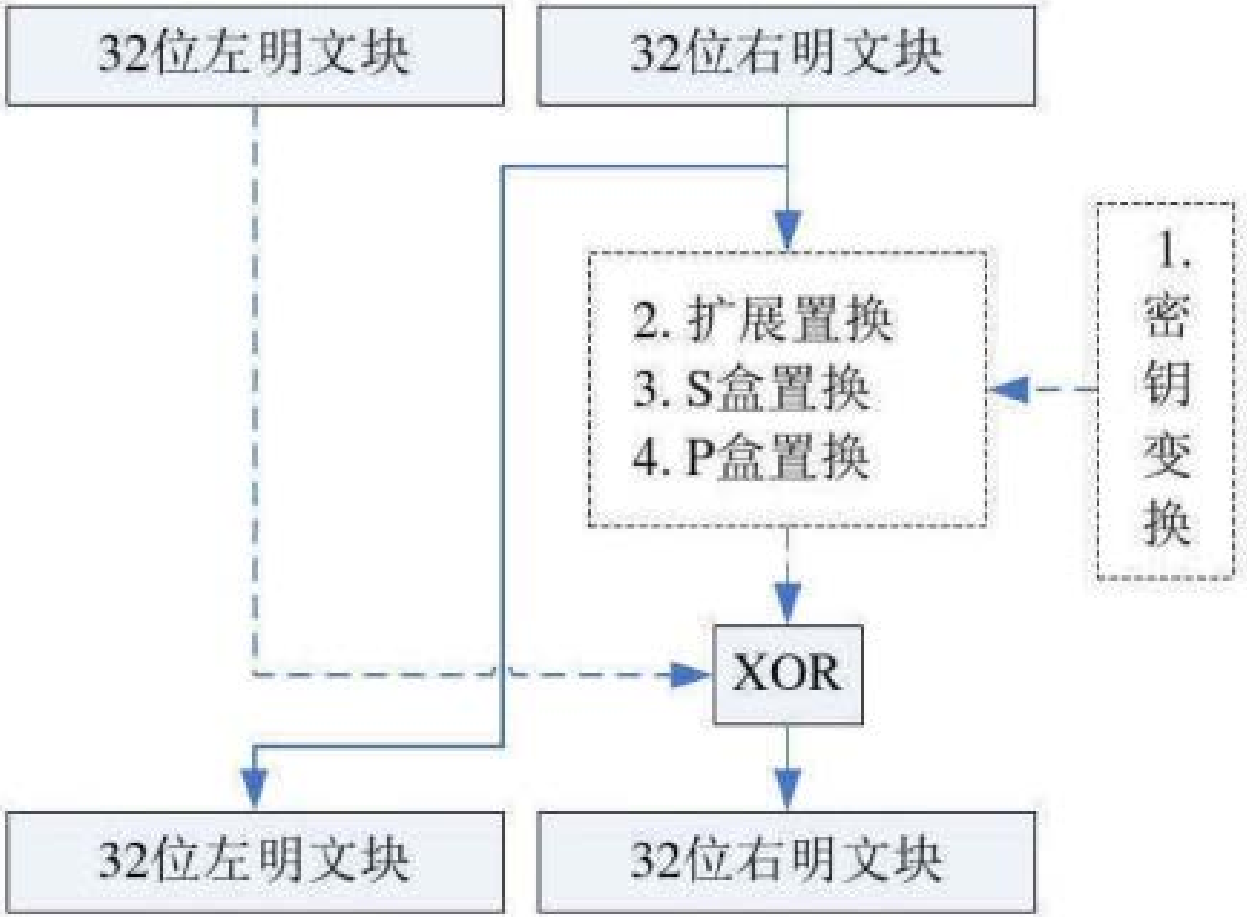
密钥的产生

14	17	11	24	1	5
3	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

密钥变换


密钥变换
扩展置换
S盒替换
P盒替换
异或与交换
48位输入块
6位子块
6位子块
… …
6位子块
S1盒
S2盒
… …
S8盒
4位输出
4位输出
… …
4位输出
32位输出块

异或与交换


32位左明文块
32位右明文块
2. 扩展置换
3. S盒置换
4. P盒置换
1. 密钥变换
XOR
32位左明文块
32位右明文块

4.3 DES例子与强度

DES的强度

DES具有很好的雪崩效应

- 雪崩效应——明文或密钥某一位发生变化（微小的变化）将对密文产生很大的影响。
- 密钥的使用（密钥长度问题）
- S盒的问题
- 计时攻击问题
- 差分分析和线性分析

密钥长度问题

- 关于DES算法的一个最有争议的问题就是担心实际56比特的密钥长度不足以抵御穷举式攻击，因为密钥量只有$2^{56} \approx 10^{17}$个
- 早在1977年，Diffie和Hellman已建议制造一个每秒能测试100万个密钥的VLSI芯片。每秒测试100万个密钥的机器大约需要一天就可以搜索整个密钥空间。他们估计制造这样的机器大约需要2000万美元。

密钥长度问题

- 在CRYPTO’93上，Session和Wiener给出了一个非常详细的密钥搜索机器的设计方案，这个机器基于并行运算的密钥搜索芯片，所以16次加密能同时完成。此芯片每秒能测试5000万个密钥，用5760个芯片组成的系统需要花费10万美元，它平均用1.5天左右就可找到DES密钥。
- 1997年1月28日，美国的RSA数据安全公司在RSA安全年会上公布了一项“秘密密钥挑战”竞赛，其中包括悬赏1万美元破译密钥长度为56比特的DES。美国克罗拉多洲的程序员Verser从1997年2月18日起，用了96天时间，在Internet上数万名志愿者的协同工作下，成功地找到了DES的密钥，赢得了悬赏的1万美元。

密钥长度问题

- 1998年7月电子前沿基金会（EFF）使用一台25万美元的电脑在56小时内破译了56比特密钥的DES。

- 1999年1月RSA数据安全会议期间，电子前沿基金会用22小时15分钟就宣告破解了一个DES的密钥。

S盒的问题

- F函数(S-Box)设计原理未知
- 1976年美国NSA提出了下列几条S盒的设计准则：

1. S盒的每一行是整数0，…，15的一个置换
2. 没有一个S盒是它输入变量的线性函数
3. 改变S盒的一个输入位至少要引起两位的输出改变
4. 对任何一个S盒和任何一个输入X，S（X）和S(X⊕001100）至少有两个比特不同（这里X是长度为6的比特串）
5. 对任何一个S盒，对任何一个输入对e,f属于{0,1}，

$$S(X) \neq S(X \oplus 11ef00)$$

6. 对任何一个S盒，如果固定一个输入比特，来看一个固定输出比特的值，这个输出比特为0的输入数目将接近于这个输出比特为1的输入数目。

计时攻击

- 通过对执行给定的多种密文解密所需时间的观察，来获得关于密钥或明文的信息。
- 利用加/解密算法对于不同的输入所花的时间有着细微的差别。

DES可以抵御计时攻击。

差分分析和线性分析

- 1990年，以色列密码学家Eli Biham和Adi Shamir提出了差分密码分析法，可对DES进行选择明文攻击。

- 线性密码分析比差分密码分析更有效。

弱密钥与半弱密钥

- 弱密钥：$E_K \bullet E_K = I$，DES存在8个弱密钥

$$E_k(E_k(m)) = m$$

0x0101010101010101　　0x0000000000000000

0xFEFEFEFEFEFEFEFE　　0xFFFFFFFFFFFFFFFF

0xE0E0E0E0F1F1F1F1　　0xE1E1E1E1F0F0F0F0

0x1F1F1F1F0E0E0E0E　　0x1E1E1E1E0F0F0F0F

弱密钥与半弱密钥

- 半弱密钥: $E_{K0} = E_{K1}$，至少有12个半弱密钥

$$E_{k_1}(E_{k_0}(m)) = m$$

- 0x011F011F010E010E 和 0x1F011F010E010E01
- 0x01E001E001F101F1 和 0xE001E001F101F101
- 0x01FE01FE01FE01FE 和 0xFE01FE01FE01FE01
- 0x1FE01FE00EF10EF1 和 0xE01FE01FF10EF10E
- 0x1FFE1FFE0EFE0EFE 和 0xFE1FFE1FFE0EFE0E
- 0xE0FEE0FEF1FEF1FE 和 0xFEE0FEE0FEF1FEF1

4.4 分组密码的设计原理

分组密码的设计原理

- DES的设计标准
  - S盒的设计准则
  - 置换P的设计准则
- 迭代轮数
- 轮函数F的设计
  - 严格雪崩效应准则
  - 独立准则
- 密钥扩展算法

分组密码的设计原理

- 严格雪崩效应准则（SAC）（明文或密钥的某一位发生变化会导致密文的很多位发生变化）
  - S盒的输入的任意一位i发生变化，输出的任意一位j发生变化的可能性为1/2。
  - 增强扩散特性。扩散：明文和密文间关系为非线性关系
  - 独立准则（BIC）
  - 对任意的i、j、k，当输入中的一位i发生变换时，输出位中的j和k位的变化是彼此无关的。
  - 加强混淆的有效性。混淆：密钥和密文关系为非线性关系

实现的设计原则

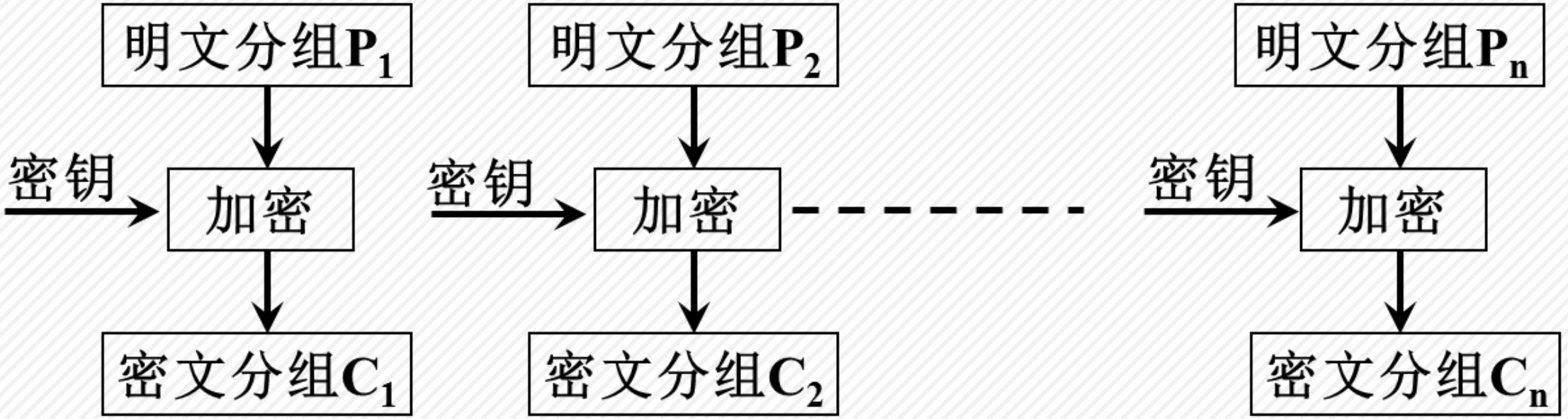
- 软件实现的要求：使用子块和简单的运算。密码运算在子块上进行，要求子块的长度能自然地适应软件编程，如8、16、32比特等。应尽量避免按比特置换，在子块上所进行的密码运算尽量采用易于软件实现的运算。最好是用处理器的基本运算，如加法、乘法、移位等。
- 硬件实现的要求：加密和解密的相似性，即加密和解密过程的不同仅仅在密钥使用方式上，以便采用同样的器件来实现加密和解密，以节省费用和体积。尽量采用标准的组件结构，以便能适应于在超大规模集成电路中实现。

分组密码的工作模式

- 分组密码是将消息作为数据分组来加密或解密的，而实际应用中大多数消息的长度是不定的，数据格式也不同。当消息长度大于分组长度时，需要分成几个分组分别进行处理。为了能灵活地运用基本的分组密码算法，人们设计了不同的处理方式，称为**分组密码的工作模式**，也称为分组密码算法的运行模式。

- 分组模式能为密文组提供一些其他的性质，例如隐藏明文的统计特性、数据格式、控制错误传播等，以提高整体安全性，降低删除、重放、插入和伪造等攻击的机会。常用的工作模式有**电子编码本**模式、**密码分组链接**模式、**输出反馈**模式、**密码反馈**模式。

加密过程



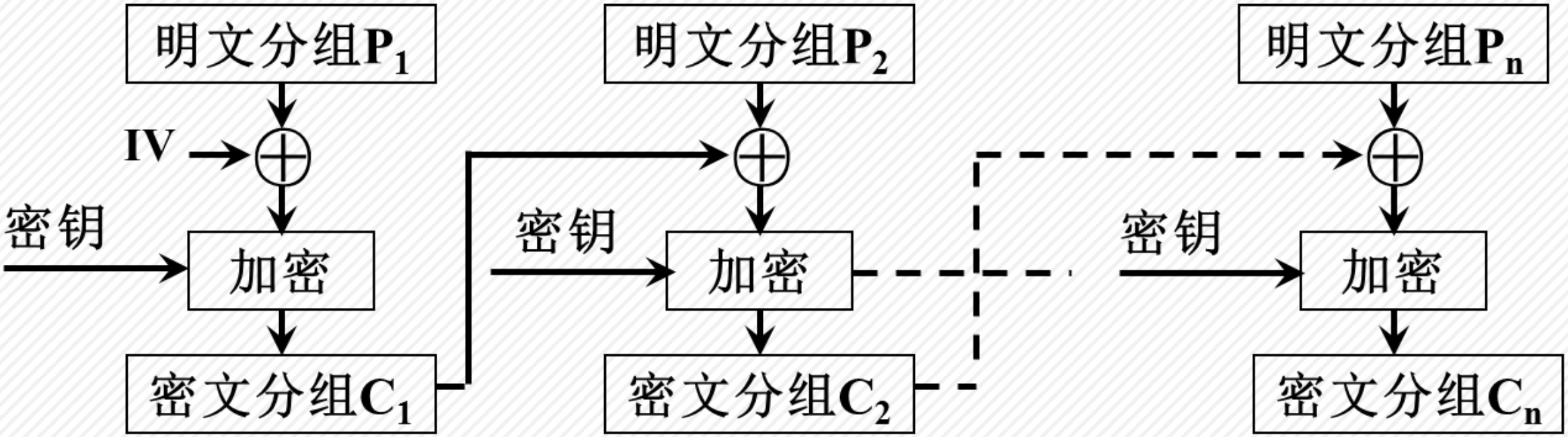
电子编码本模式(Electronic Code Book，ECB)

✓**优点:**1.简单；2.有利于并行计算；3.误差不会被传送；

✓**缺点:**1.不能隐藏明文的模式；2.可能对明文进行主动攻击；

这种方法一旦有一个块被破解，使用相同的方法可以解密所有的明文数据，安全性比较差。

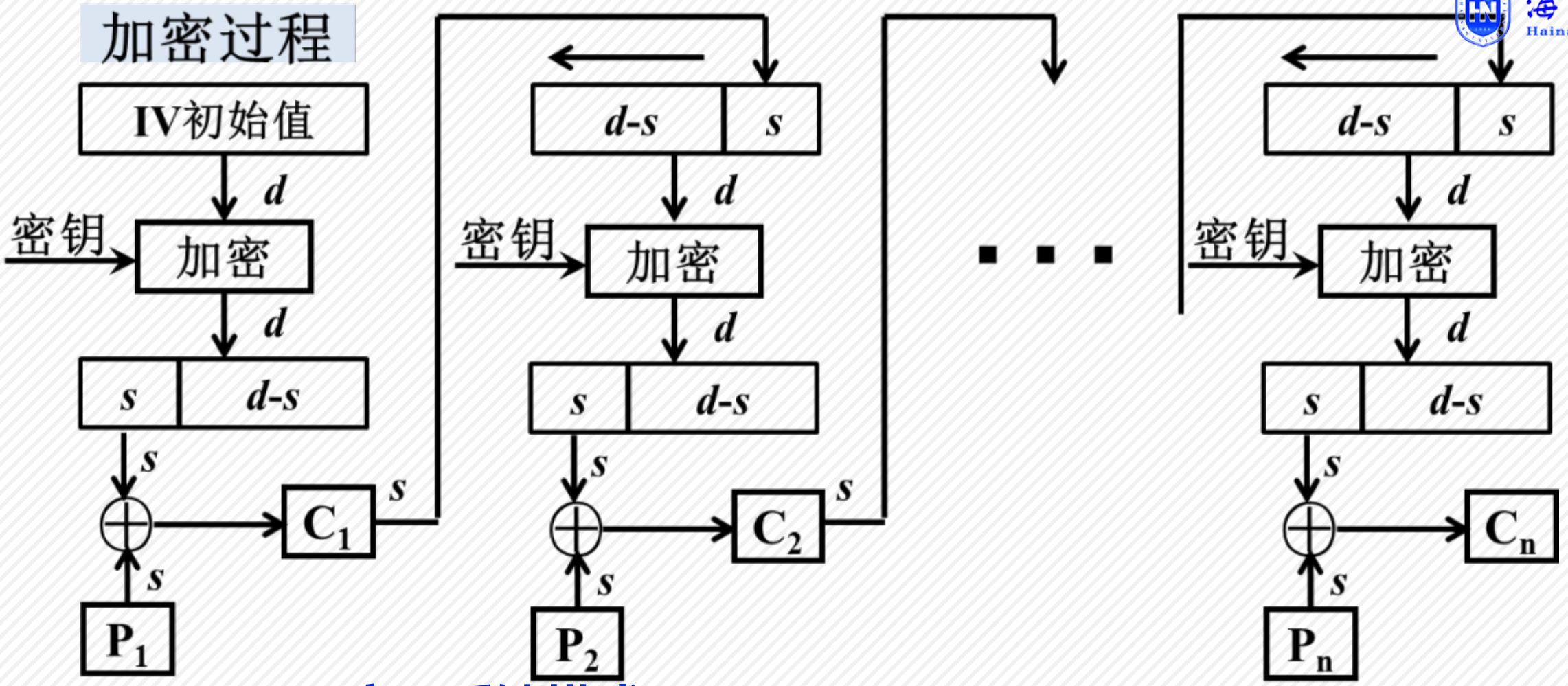
加密过程



密码分组链接模式(Cipher Block Chaining，CBC)

✓**优点：**不容易主动攻击,安全性好于ECB,适合传输长度长的报文,是SSL、IPSec的标准。

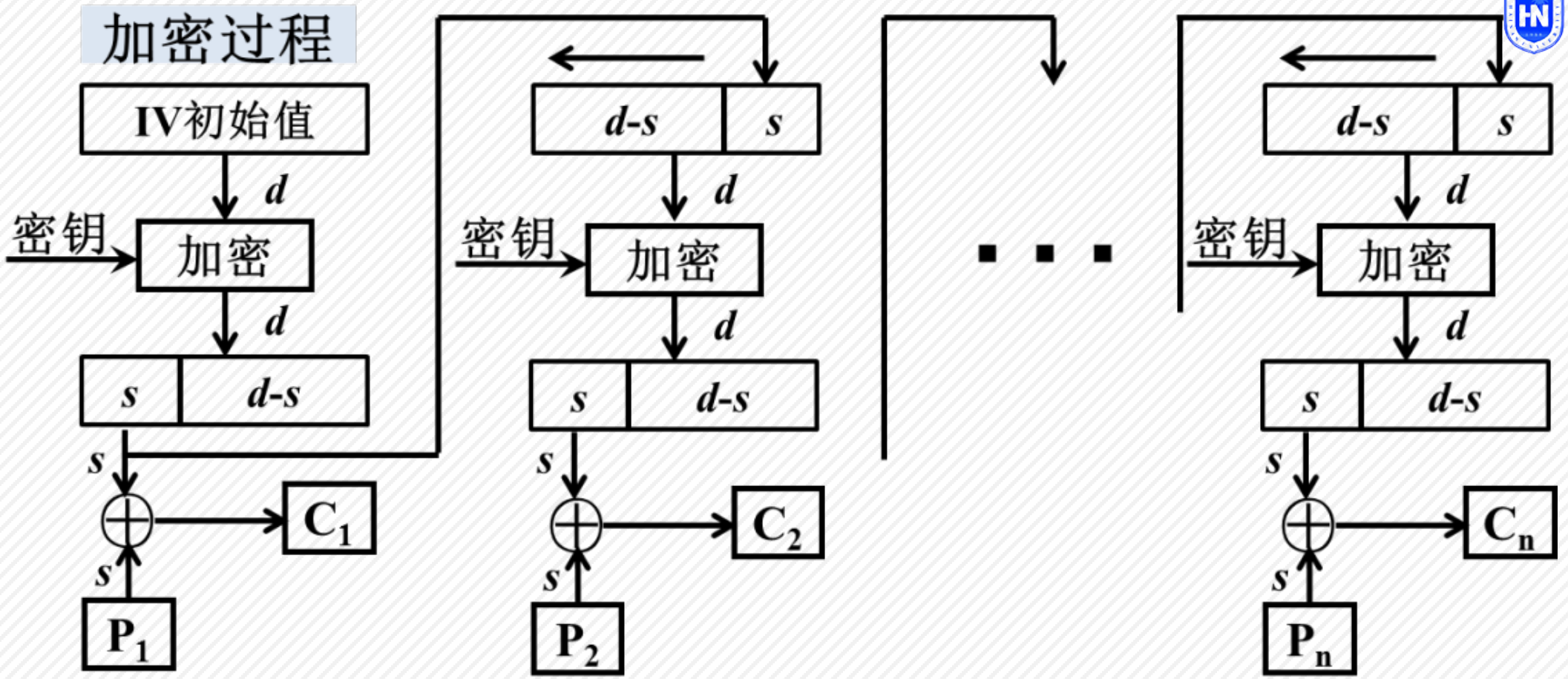
✓**缺点：**1.不利于并行计算；2.误差传递；3.需要初始化向量IV



密码反馈模式 (Cipher FeedBack，CFB)

✓ **优点：** 1.隐藏了明文模式;2.分组密码转化为流模式;3.可以及时加密传送小于分组的数据;

✓**缺点:**1.不利于并行计算;2.误差传送：一个明文单元损坏影响多个单元;3.唯一的IV;



输出反馈模式 (Output FeedBack，OFB)

- ✓ **优点:**1.隐藏了明文模式;2.分组密码转化为流模式;3.可以及时加密传送小于分组的数据;
- ✓**缺点:**1.不利于并行计算;2.对明文的主动攻击是可能的;3.误差传送：一个明文单元损坏影响多个单元;


海南大学
Hainan university