

# 密码学

## 第四章 分组密码

网络空间安全学院

朱丹

zhudan@nwpu.edu.cn

- ✎ 将明文按规定的长度**分组**
- ✎ 密文的一个比特与**整个明文分组**相关
- ✎ 实质是较**复杂的单表代替密码**

密钥  $k = (k_1, k_2, \dots, k_r)$

密钥  $k = (k_1, k_2, \dots, k_r)$

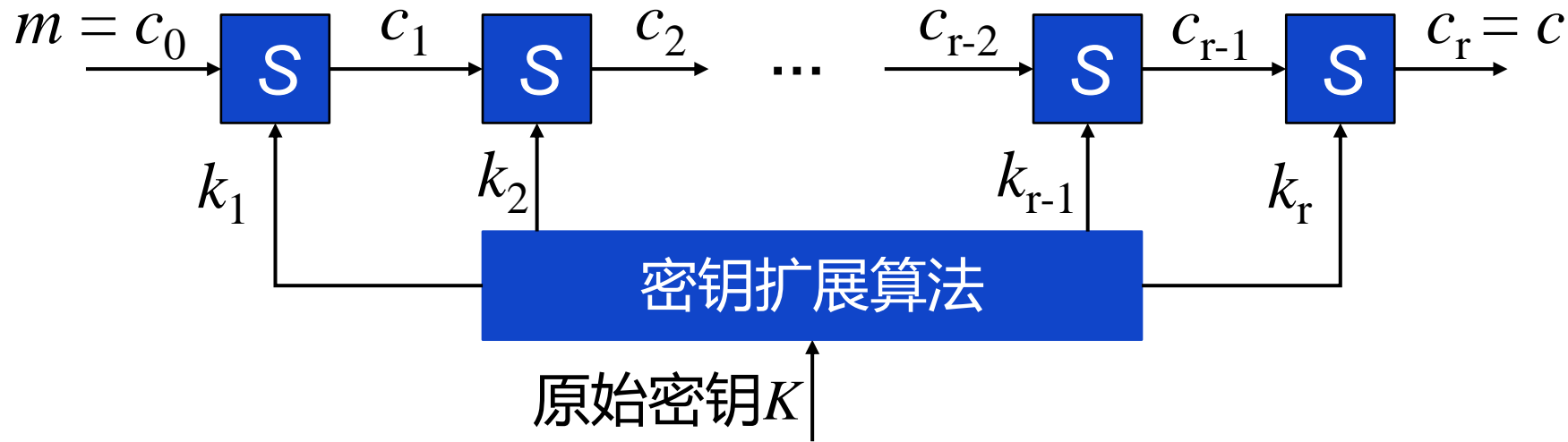
$m = (m_1, m_2, \dots, m_n)$

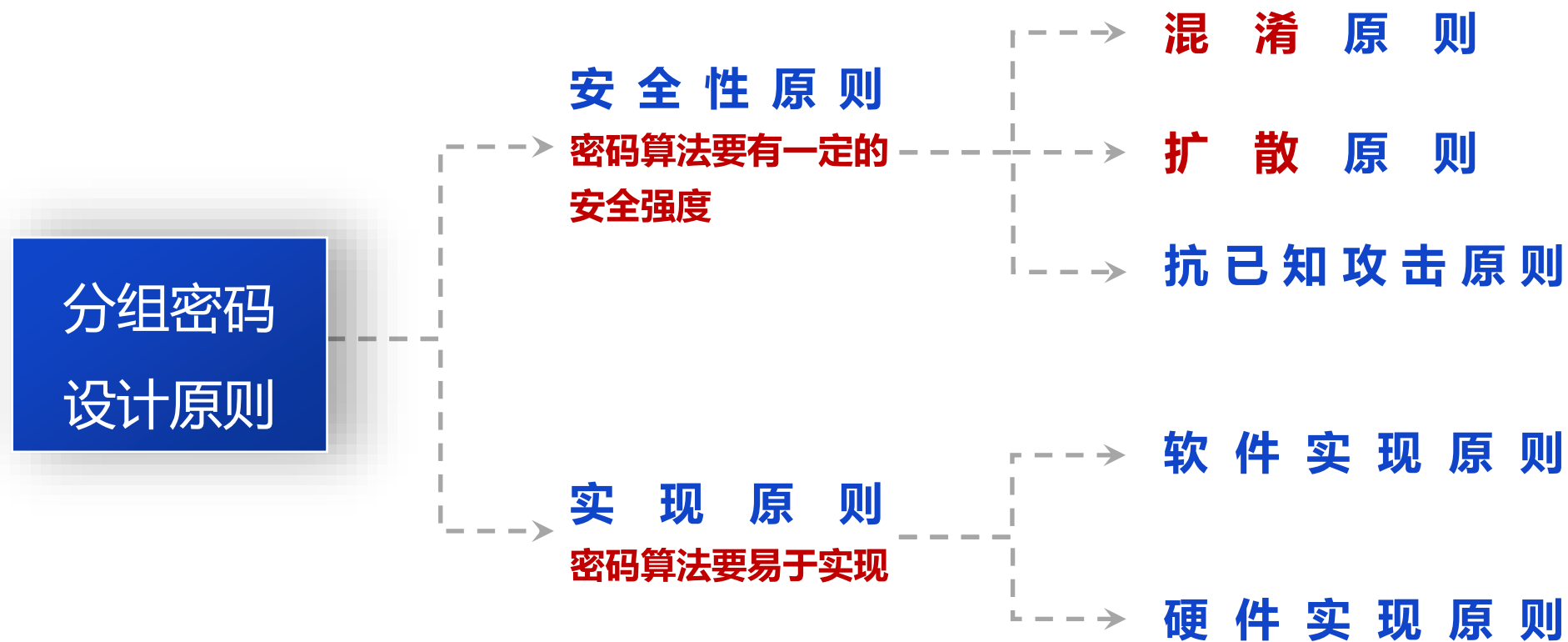
$c = (c_1, c_2, \dots, c_n)$

$m = (m_1, m_2, \dots, m_n)$

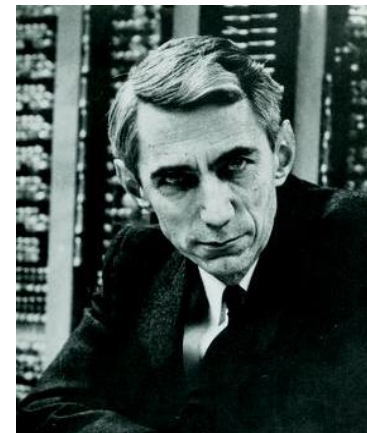


- 定义：对于非幂等的密码体制 $S$ ，将自身做 $n$ 次乘积得到的密码 $S^n$ 称为 $S$ 的 $n$ 重迭代密码
  - 迭代密码可以通过简单密码得到高强度密码
  - 迭代密码是现代分组密码和杂凑函数的核心设计思想
- 迭代型分组密码将原始密钥经密钥扩展算法得到多个轮密钥，每一轮使用一个轮密钥





- ✦ 混淆 (Confusion) 原则：要求所设计的密码应该是**密钥和密文之间的依赖关系尽可能复杂**，以至于无法被攻击者利用
- ✦ 扩散 (Diffusion) 原则：**明文的每个比特位影响密文尽可能多的比特位**，输入微小的变化导致输出多位变化



Claude E. Shannon

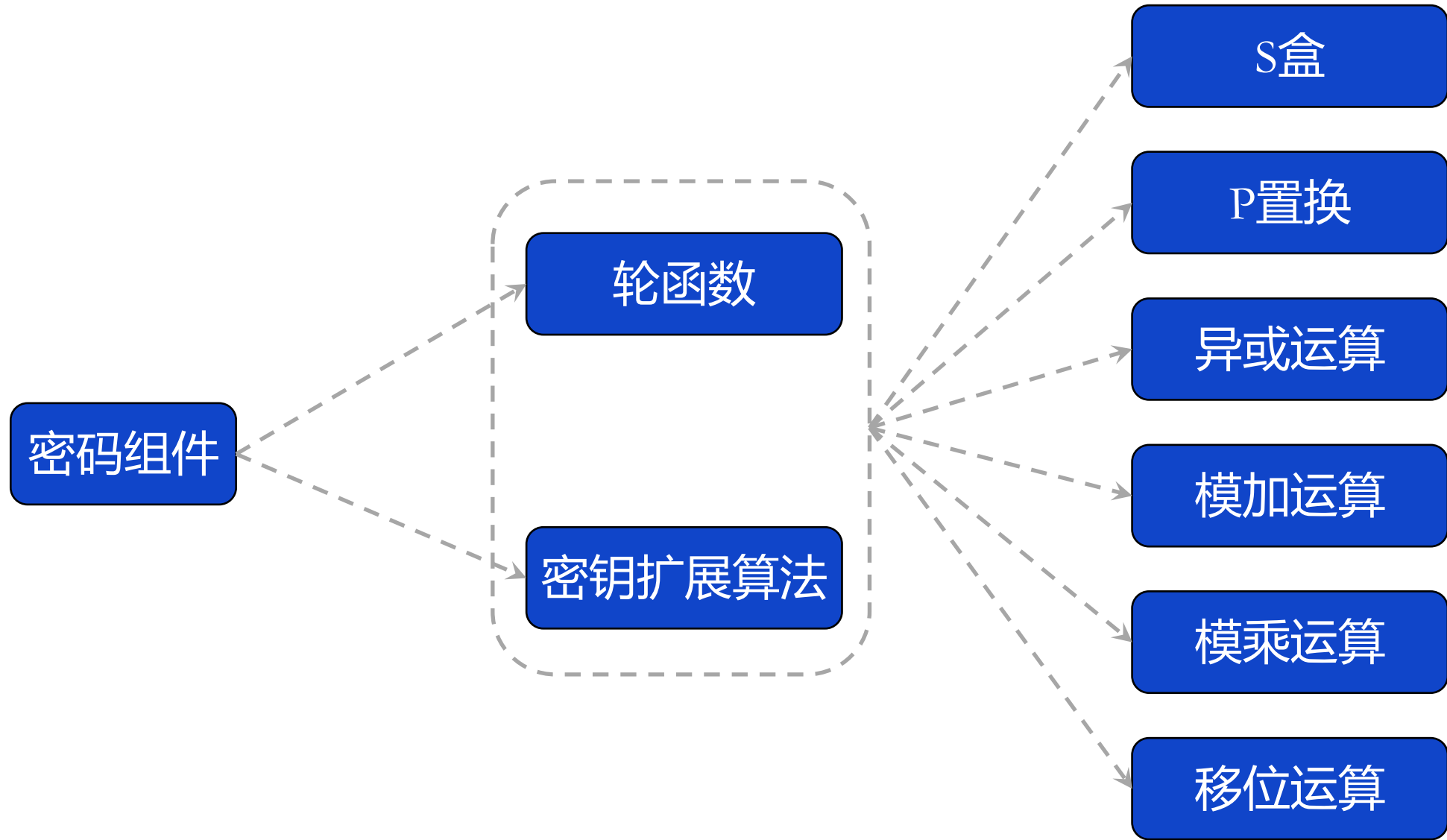
在密码学当中，**混淆** (confusion) 与**扩散** (diffusion) 是设计密码学算法的两种主要方法。这样的定义最早出现在克劳德·香农1945年的论文《**密码学的数学理论**》当中。

在克劳德·香农的定义之中，混淆主要是用来使密文和对称式加密方法中密钥的关系变得尽可能的复杂；扩散则主要是用来使用明文和密文的关系变得尽可能的复杂，明文中任何一点小更动都会使得密文有很大的差异。

**混淆**用于掩盖**密钥与密文之间的关系**。这可以挫败通过研究密文以获取冗余度和统计模式的企图。做到这一点最容易的方法是“**代替**”。

**扩散**通过将明文冗余度分散到密文中使之分散开来。即将**单个明文比特的影响尽可能扩大到更多的密文比特中去**。产生扩散最简单的方法是**换位 (置换)**。

- ✦ 理解混淆和扩散的含义，准确掌握混淆和扩散描述的是谁和谁之间的关系
- ✦ 在唯密文攻击模型下：掌握密文能够获得尽量少的关于密钥和明文的信息，通过互信息来度量





加密模式		特点
Electronic Code Book(ECB)	电子密码本模式	简单快速，可并行计算
Cipher Block Chaining(CBC)	密码分组链接模式	仅解密支持并行计算
Cipher Feedback Mode(CFB)	密文反馈模式	仅解密支持并行计算
Output Feedback Mode(OFB)	输出反馈模式	不支持并行运算
Counter (CTR)	计数器模式	支持并行计算



# 章节安排

Outline



DES算法概述

---



DES加解密算法

---



DES密钥扩展算法

---



DES安全性分析

---

# 章节安排

Outline



DES算法概述

---



DES加解密算法

---



DES密钥扩展算法

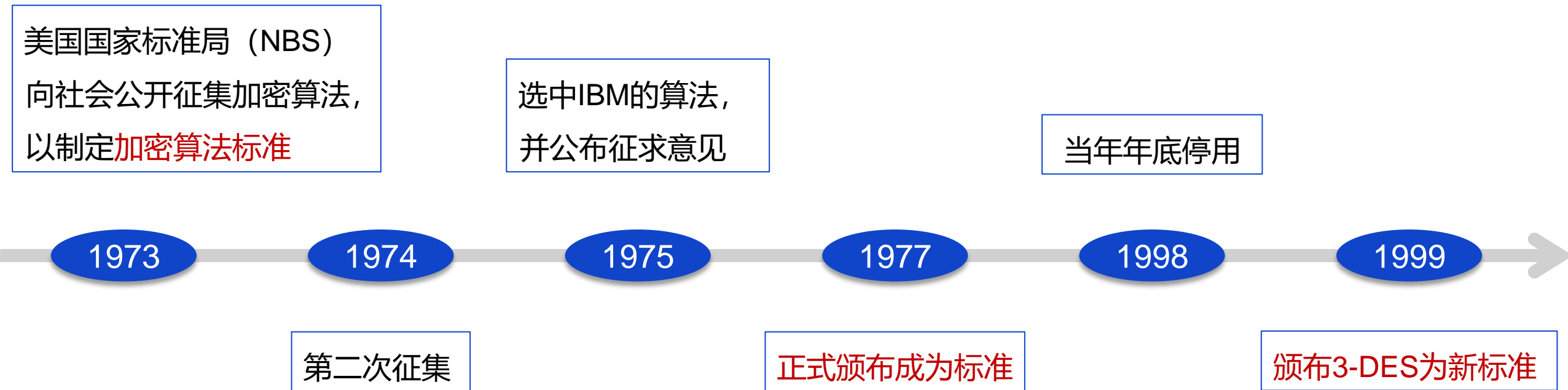
---



DES安全性分析

---

### ✎ DES – Data Encryption Standard (数据加密标准)






- ✎ DES是公开征集的算法标准, 算法是公开的
- ✎ DES算法启用是现代密码算法发展的两大代表性事件之一, 另一个是公钥密码概念的提出




- ✎ 标准加密算法的设计目标
  - ✎ 用于加密保护政府机构和商业部门的**非机密的敏感数据**
  - ✎ 用于加密保护**静态存储**和**传输信道**中的数据
  - ✎ 设计时预期安全使用**10 ~ 15年**
  - ✎ DES使用中每**5年**评估一次

- ✎ 分组密码：明文、密文和密钥的**分组长度**都是**64位**
- ✎ 综合运用了**置换、代替、代数**等基本密码技术
- ✎ **基本结构属于Feistel结构**(Horst Feistel最早提出)
- ✎ **对合运算**：
  - ✎  $f = f^{-1}$
  - ✎ **加解密共用同一算法**，使工程实现的工作量减半
- ✎ 面向**二进制数据**的密码算法：适于计算机实现

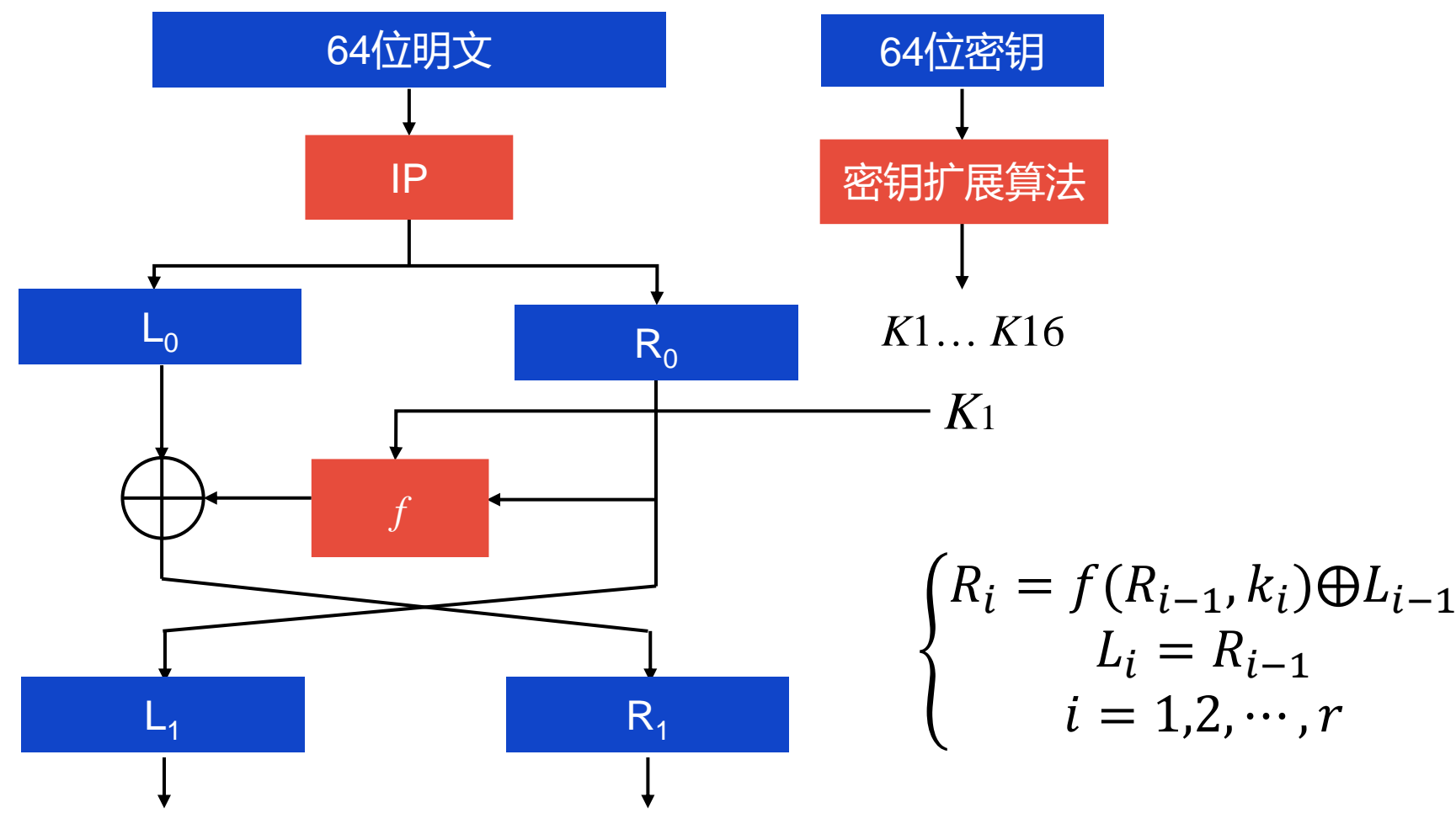
### 应用

-  曾在全世界范围得到广泛应用
-  曾被许多国际组织采用为**标准**
-  产品形式：软件（嵌入式软件，应用软件）  
硬件（芯片，智能卡，专用设备）

### 结论

-  **用于其设计目标是安全的**
-  设计精巧、实现容易、使用方便，**堪称典范**
-  为国际信息安全发挥了重要作用

✎ Feistel结构



- ✎ 掌握DES的整体算法结构：分组长度、IP、轮函数、IP<sup>-1</sup>、密钥扩展
- ✎ 掌握DES算法的Feistel网络结构、16轮的迭代结构；与迭代函数的概念结合起来

## Structure

The diagram illustrates the decryption process for a 64-bit block cipher. The process begins with a 64-bit ciphertext input, which is processed by an inverse initial permutation ( $IP^{-1}$ ) block. The output is split into two 32-bit halves,  $L_{16}$  and  $R_{16}$ . These halves pass through 16 rounds of decryption. Each round consists of a function block  $f$  (red) and a rotation block (blue). The function block  $f$  takes the right half  $R_i$  and a round key  $K_i$  as input and outputs the result to the left half  $L_i$ . The rotation block then swaps the two halves. The final output is the 64-bit plaintext.

✎ 掌握DES算法的Feistel网络结构、16轮的迭代结构；与迭代函数的概念结合起来



# 章节安排

Outline



DES算法概述

---



DES加解密算法

---



DES密钥扩展算法

---



DES安全性分析

---

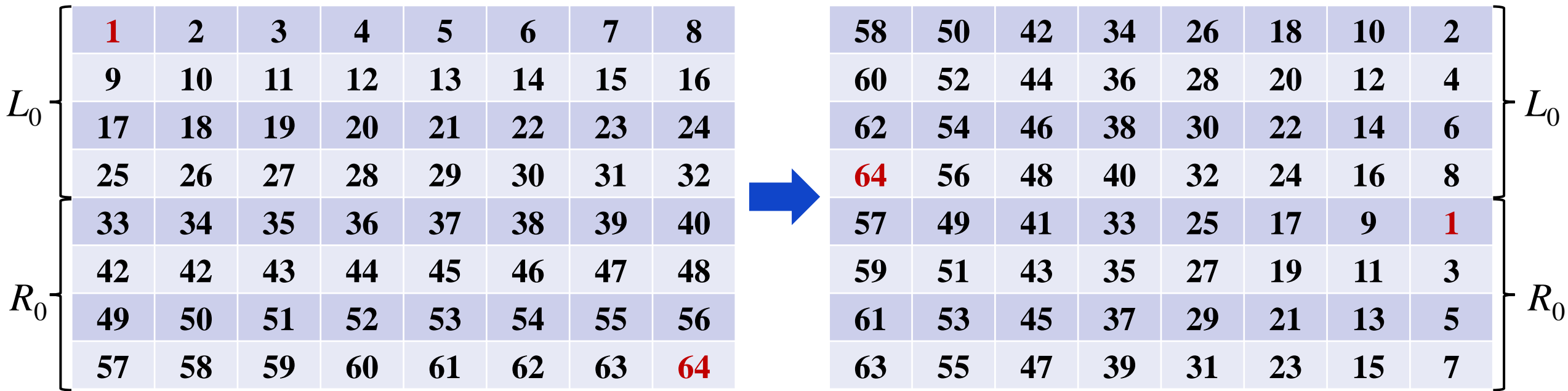
- ✎ 64位密钥经子密钥产生算法产生出16个**子密钥**：  $K_1, K_2, \dots, K_{16}$ ，分别供16轮加密迭代使用
- ✎ 64位明文经初始置换IP，将数据打乱重排并分成左右两半。左边为  $L_0$ ，右边为  $R_0$
- ✎ 第一次加密迭代：
  - ✎ 在子密钥  $K_1$  的控制下，由加密函数  $f$  对  $R_0$  加密
$$L_0 \oplus f(R_0, K_1)$$
  - ✎ 以此作为第二次加密迭代的  $R_1$ ，以  $R_0$  作为第二次加密迭代的  $L_1$

- ✎ 第二次加密迭代至第16次加密迭代分别用子密钥 $K_2, \dots, K_{16}$ 进行, 其过程与第一次加密迭代相同
- ✎ 第16次加密迭代结束后, 产生一个64位的数据组。以其左边32位作为 $L_{16}$ , 以其右边32位作为 $R_{16}$
- ✎  $R_{16}$ 与 $L_{16}$ 合并, 再经过逆初始置换 $IP^{-1}$ , 将数据重新排列, 便得到64位密文

$$\begin{cases} L_i = R_{i-1} \\ R_i = L_{i-1} \oplus f(R_{i-1}, K_i) \\ i = 1, 2, \dots, 16 \end{cases}$$

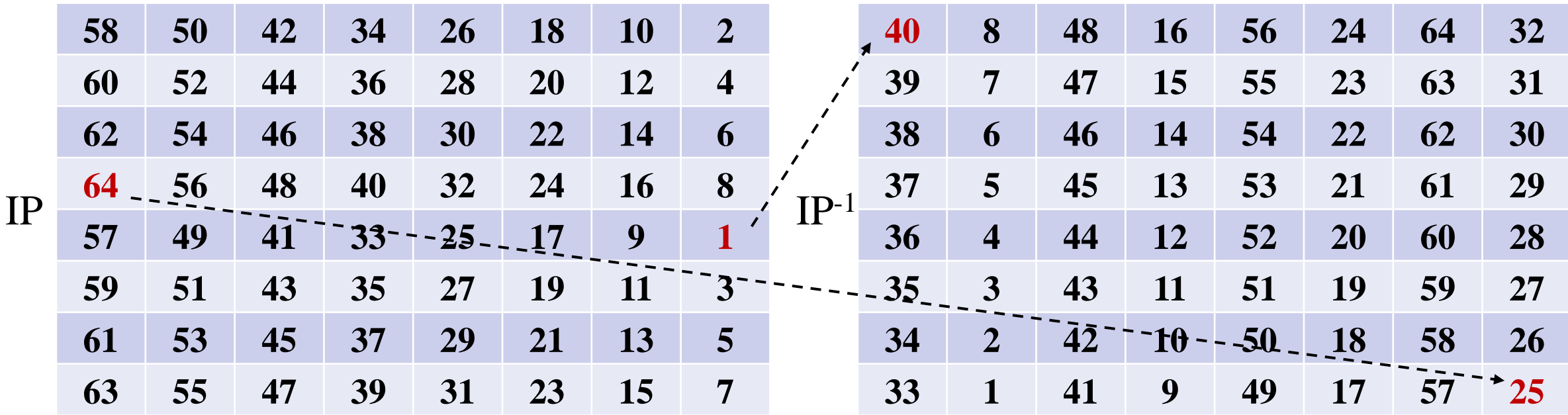
- ✎ 初始置换IP: 把64位明文打乱重排
- ✎ 左一半为 $L_0$ (左32位), 右一半为 $R_0$ (右32位)

置换矩阵具有明显的规律, 这对安全性是不利的



- ✎ IP置换的本质: 比特级别的置换, 置乱
- ✎ IP对安全性的贡献: 置乱, 打破明文的跟随关系, 但是对提升安全性意义不大

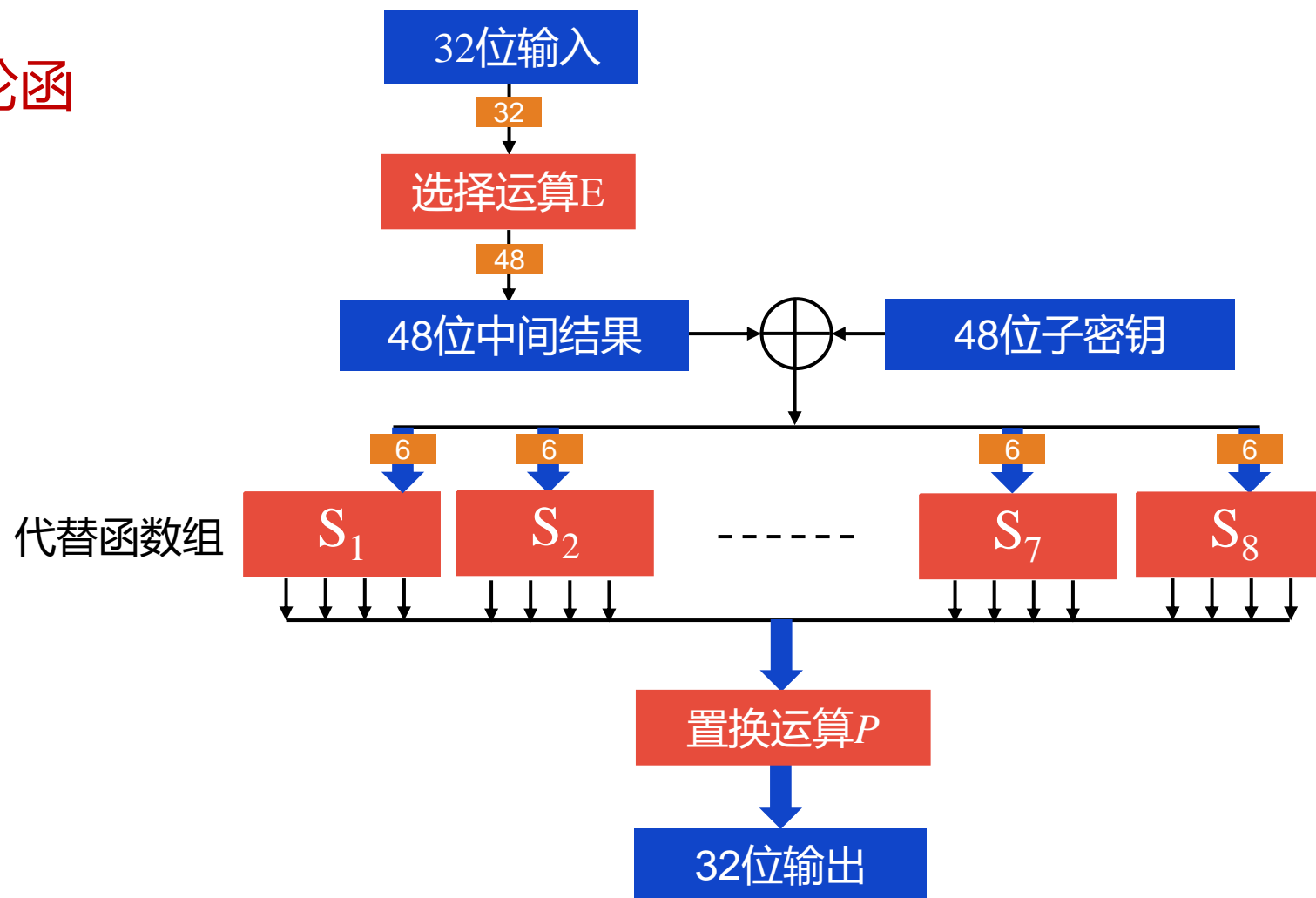
- ✎ 逆初始置换IP<sup>-1</sup>: IP与IP<sup>-1</sup>互逆
- ✎ 例: 在IP中把输入的第1位置换到第40位, 而在IP<sup>-1</sup>中把输入的第40位置换回第1位
- ✎ 保密作用不大: 没有密钥参与, IP和IP<sup>-1</sup>均公开, 保密意义不大



✎ IP置换的本质: 比特级别的置换, 置乱

✎ IP对安全性的贡献: 置乱, 打破明文的跟随关系, 但是对提升安全性意义不大

- 加密函数  $f$ : DES的轮函数, DES保密的核心



- ✎ 选择运算E：把32位输入扩充为48位中间数据
- ✎ 通过重复使用数据，实现数据扩充
- ✎ 选择矩阵

32	1	2	3	4	5	4	5	6	7	8	9
8	9	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	1

- ✎ 选择运算E的本质：比特级扩充，为了和轮密钥的宽度（48位）相匹配
- ✎ 从信息论的角度看，48比特的选择运算结果有冗余

### ✎ DES代替函数组S (S盒)

- ✎ S盒是DES中唯一的非线性变换，是DES安全的关键
- ✎ 在保密性方面，起混淆作用
- ✎ 共有8个S盒，并行工作
- ✎ 每个S盒有6个输入，4个输出，是非线性压缩变换
- ✎ 设输入为 $b_1b_2b_3b_4b_5b_6$ ，则以 $b_1b_6$ 组成的二进制数为行号， $b_2b_3b_4b_5$ 组成的二进制数为列号，行列交点处的数为输出



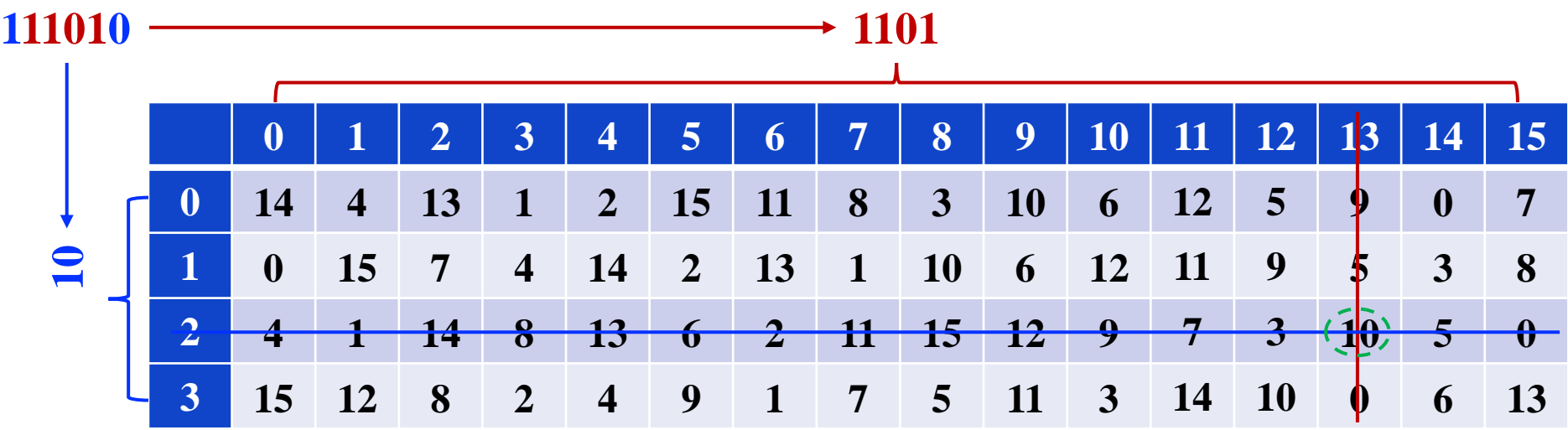
## DES代替函数组S (S盒)

- 每个S盒有6个输入，4个输出，是非线性压缩变换
- 设输入为 $b_1b_2b_3b_4b_5b_6$ ，则以 $b_1b_6$ 组成的二进制数为行号， $b_2b_3b_4b_5$ 组成的二进制数为列号，行列交点处的数为输出

		$b_2b_3b_4b_5$															
		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$b_1b_6$	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

## DES代替函数组S (S盒)

- 每个S盒有6个输入，4个输出，是非线性压缩变换
- 设输入为 $b_1b_2b_3b_4b_5b_6$ ，则以 $b_1b_6$ 组成的二进制数为行号， $b_2b_3b_4b_5$ 组成的二进制数为列号，行列交点处的数为输出



✎ 1976年, NSA公布的DES的S盒设计准则:

- ✎ P0: 每个S盒的每一行都是整数0到15的一个置换
- ✎ P1: 每个S盒的输出不是它的输入的线性或仿射函数
- ✎ P2: 改变S盒的任一输入比特, 其输出至少有两比特发生改变
- ✎ P3: 对任一S盒和任一输入 $x$ ,  $S(x)$ 和 $S(x \oplus 001100)$ 至少有两位发生变化
- ✎ P4: 对任何S盒和任一输入 $x$ , 以及 $e, f \in \{0, 1\}$ , 有 $S(x) \neq S(x \oplus 11ef00)$
- ✎ P5: 对任何S盒, 当它的任一输入比特位保持不变, 其它5位改变时, 输出数字中0和1的数目大致相等

✎ S盒的设计规则, 每一行都是0到15的一个置换, 从16的全排列种可能性种选出了32行作为8个S盒设计

## 4.4(1) DES加密算法

### ✎ S盒设计准则验证

- ✎ P2: 改变S盒的任一输入比特, 其输出至少有两比特发生改变
- ✎ P3: 对任一S盒和任一输入 $x$ ,  $S(x)$ 和 $S(x \oplus 001100)$ 至少有两位发生变化
- ✎ P4: 对任何S盒和任一输入 $x$ , 以及 $e, f \in \{0, 1\}$ , 有 $S(x) \neq S(x \oplus 11ef00)$
- ✎ P5: 对任何S盒, 当它的任一输入比特位保持不变, 其它5位改变时, 输出数字中0和1的数目大致相等

	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

- ✎ 美国NSA至今没有完全公布S盒的设计细节。研究表明，S盒还有一些其它设计准则：
  - ✎ 非线性度准则：S盒必须有足够的非线性度，否则不能抵抗线性攻击
  - ✎ 差分均匀性准则：S盒的差分性应均匀，否则不能抵抗差分攻击
  - ✎ 代数次数及项数分布准则：S盒必须有足够的代数次数和项数，否则不能抵抗插值攻击和高阶差分攻击

**S盒的密码学特性对DES的安全性至关重要！**

✎ 置换运算P：把数据打乱重排，在保密性方面，起扩散作用：

✎ 因为S盒是6位输入，4位输出，其非线性作用是局部的

✎ 因此，需要把S盒的混淆作用扩散开来

✎ S盒与P置换的互相配合，共同确保DES的安全

✎ 置换矩阵：

16	7	20	21	29	12	28	17
1	15	23	26	5	18	31	10
2	8	24	14	32	27	3	9
19	13	30	6	22	11	4	25

✎ 结合DES算法实例理解S盒和P盒结合使用的设计思想

✎ S盒和P盒分别实现什么？结合起来达到的效果

- DES的加密算法是**对合运算**，因此解密和加密可共用同一个算法
- 不同之处：**子密钥使用的顺序不同**
  - 第一次解密迭代使用子密钥 $K_{16}$
  - 第二次解密迭代使用子密钥 $K_{15}$
  - 第十六次解密迭代使用子密钥 $K_1$
- DES加解密过程的数学描述：

$$\begin{cases} L_i = R_{i-1} \\ R_i = L_{i-1} \oplus f(R_{i-1}, K_i) \\ i = 1, 2, \dots, 16 \end{cases}$$

$$\begin{cases} R_{i-1} = L_i \\ L_{i-1} = R_i \oplus f(L_i, K_i) \\ i = 16, 15, 14, \dots, 1 \end{cases}$$

# 章节安排

Outline



DES算法概述

---



DES加解密算法

---



DES密钥扩展算法

---



DES安全性分析

---



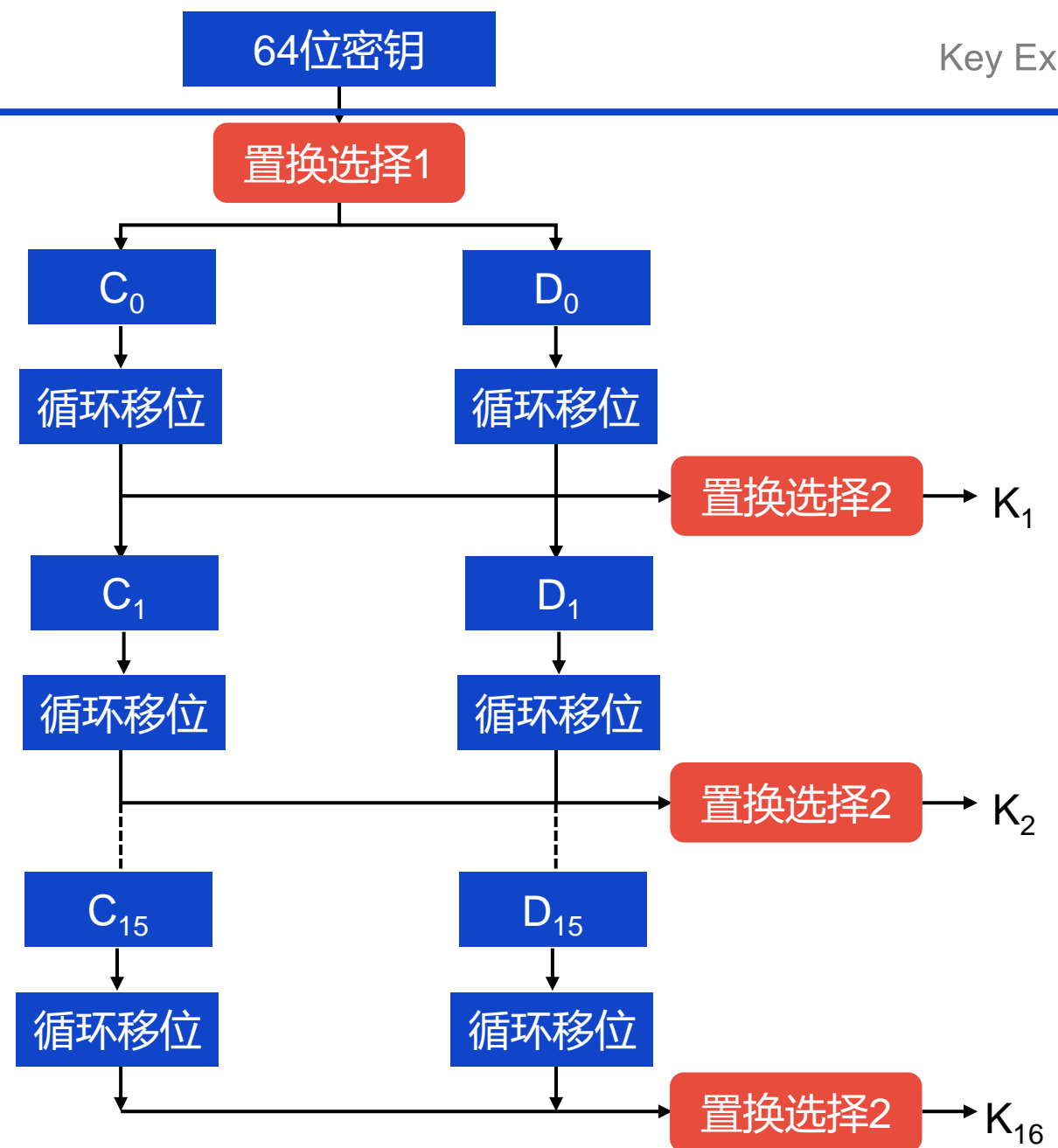
- 64位密钥经过置换选择1、循环左移、置换选择2等变换产生出16个子密钥：  $K_1, K_2, \dots, K_{16}$ ，分别供16轮加密迭代使用

$$\begin{cases} L_i = R_{i-1} \\ R_i = L_{i-1} \oplus f(R_{i-1}, K_i) \\ i = 1, 2, \dots, 16 \end{cases}$$

## 4.5 DES密钥扩展

Key Expansion

✎ 64位密钥经过置换选择1、循环左移、置换选择2等变换产生出16个子密钥：  
 $K_1, K_2, \dots, K_{16}$



✎ DES密钥扩展的整体结构：置换选择1、循环左移、置换选择2

✎ 置换选择1

- ✎ 去掉密钥中的8个奇偶校验位 (8、16、24、32、40、48、58、64)  
(有效密钥长度56位)
- ✎ 打乱重排，形成C<sub>0</sub>(左28位)， D<sub>0</sub>(右28位)

✎ 置换矩阵

C <sub>0</sub>							D <sub>0</sub>						
57	49	41	33	25	17	9	63	55	47	39	31	23	15
1	58	50	42	34	26	18	7	62	54	46	38	30	22
10	2	59	51	43	35	27	14	6	61	53	45	37	29
19	11	3	60	52	44	36	21	13	5	28	20	12	4

例，矩阵中第一个数字57，表明原密钥中的第57位移到C<sub>0</sub>中的第一位

✎ 循环移位：对 $C_0$ ,  $D_0$ 分别循环左移位

✎ 循环移位表

迭代次数	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
移位次数	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

- ✎ 置换选择2
  - ✎ 从  $C_i$  和  $D_i$  (56位)中选出48位的子密钥  $K_i$

✎ 置换矩阵

$K_i$						
14	17	11	24	1	5	选自 $C_i$
3	28	15	6	21	10	
23	19	12	4	26	8	
16	7	27	20	13	2	
41	52	31	37	47	55	选自 $D_i$
30	40	51	45	33	48	
44	49	39	56	34	53	
46	42	30	36	29	32	

从  $C_i$  中取出24位，从  $D_i$  中取出24位，形成48位的子密钥  $K_i$

# 章节安排

Outline



DES算法概述

---



DES加解密算法

---



DES密钥扩展算法

---



DES安全性分析

---

### ✎ 攻击类型

- ✎ 穷举攻击：目前最有效的方法
- ✎ 侧信道攻击：能量分析，故障注入分析
- ✎ 差分攻击：E. Biham和A. Shamir提出
- ✎ 线性攻击：M. Matsui提出

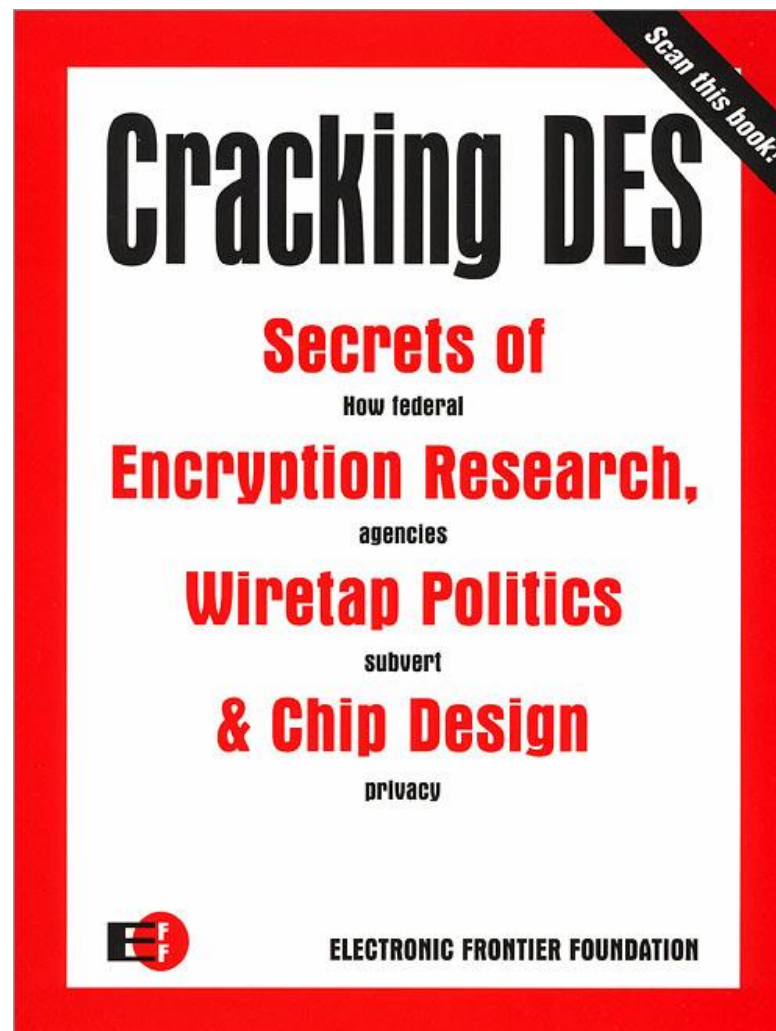
### ✎ 安全脆弱点：

- ✎ 密钥太短：有效密钥长度只有56位（64位密钥含8位奇偶校验位）
- ✎ 存在弱密钥：设 $C = \text{DES}(M, K)$ ,  $M = \text{DES}(C, K)$
- ✎ 存在互补对称性：由异或运算导致

设 $C = \text{DES}(M, K)$ , 则有 $\bar{C} = \text{DES}(\bar{M}, \bar{K})$

E.Biham A.Shamir. Differential Cryptanalysis of DES-like Cryptosystems,1999

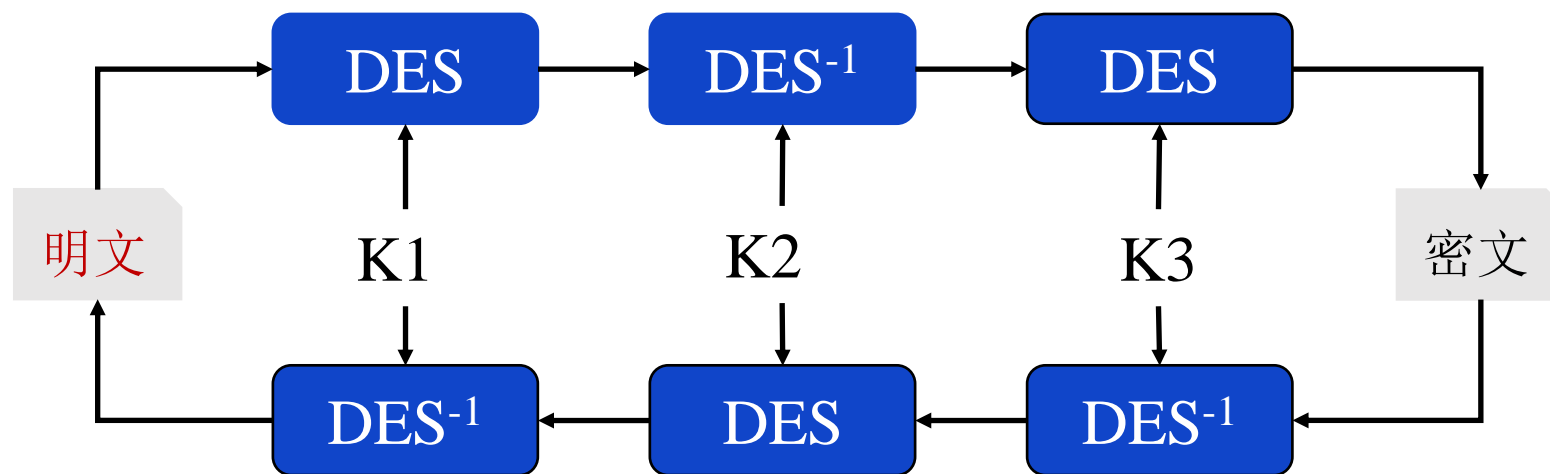
M. Matsui. Linear Cryptoanalysis Method for DES Cipher, 1993.









- ✎ 美国NIST在1999年发布了一个新版本的DES标准 (FIPS PUB46-3)
  - ✎ DES只用于遗留系统
  - ✎ **3-DES**取代DES成为新的标准
  - ✎ 国际组织和我国银行都接受3-DES
- ✎ 优点:
  - ✎ 安全: **密钥足够长 (112位或168位)**
  - ✎ 经过充分的分析和实践检验
- ✎ 缺点: **加解密速度慢**

- 采用DES算法进行三轮加密来扩展密钥长度
- 密钥长度112位、168位
  - 112位：第一重和第三重密钥相同
  - 168位：三重的密钥都不相同



## DES的贡献

-  很好地体现了Shannon的密码设计思想
-  体现了密码公开设计原则，开创了公开密码算法的先例
-  代表了当时商业密码的最高水平，是商用密码的典范
-  对确保国际信息安全和提高国际密码设计水平都发挥了重要作用

## DES给我们的启示

-  商业密码应当坚持公开设计原则
-  商业密码标准应当公布算法

 DES算法的贡献：体现了分组密码设计思想；属于一种代表性的网络结构

 密码应当坚持公开设计原则，回顾现代密码设计的安全性原则：一切密码寓于密钥之中

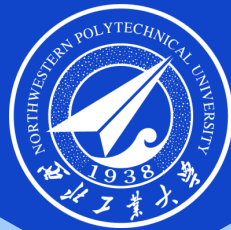
- ✎ 结合课件内容和教材深入理解DES密码算法的结构和加解密过程
- ✎ 结合课件内容和教材内容理解DES存在的安全脆弱点

```
struct Bits8
{
    unsigned b0 : 1, b1 : 1, b2 : 1, b3 : 1, b4 : 1, b5 : 1, b6 : 1, b7 : 1;
};
```

```
union CBits8
{
    struct Bits8 bits;
    unsigned char byte;
};
```

```
union byte_convert
{
    unsigned char bytes[4];
    unsigned int word;
};
```





感谢聆听!

THANK YOU FOR YOUR ATTENTION!