



密码学

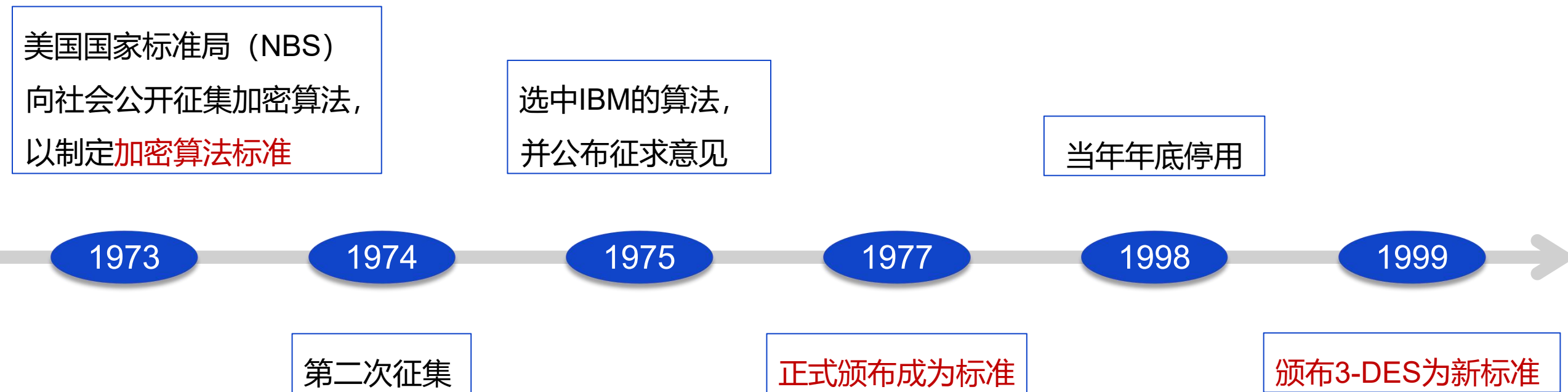
第四章 分组密码

网络空间安全学院

胡伟 朱丹

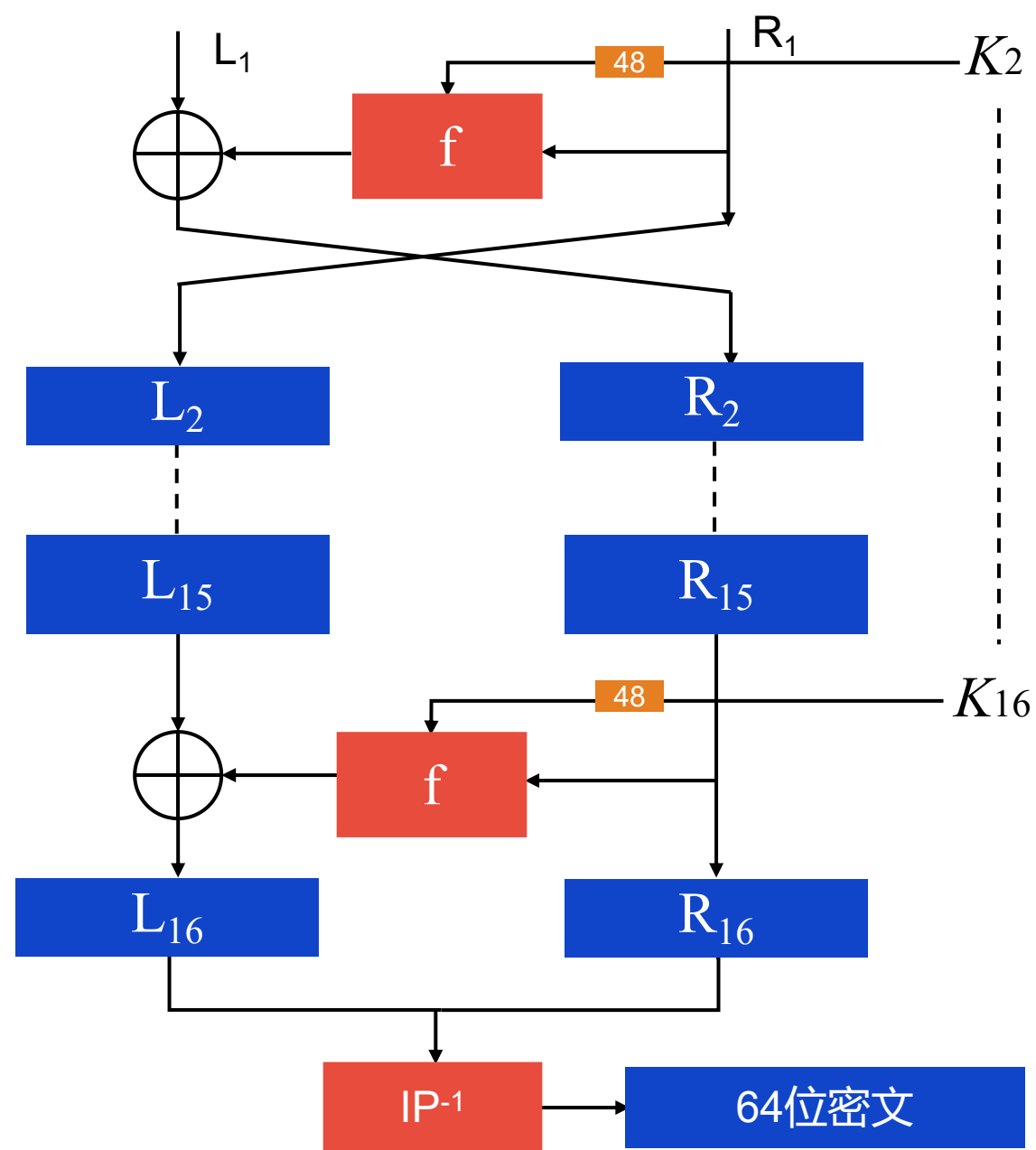
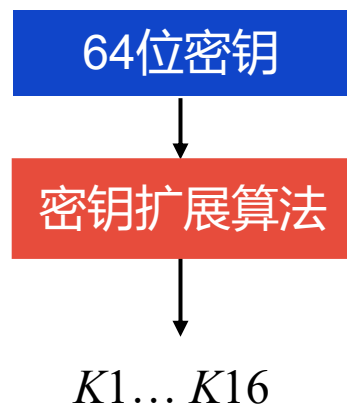
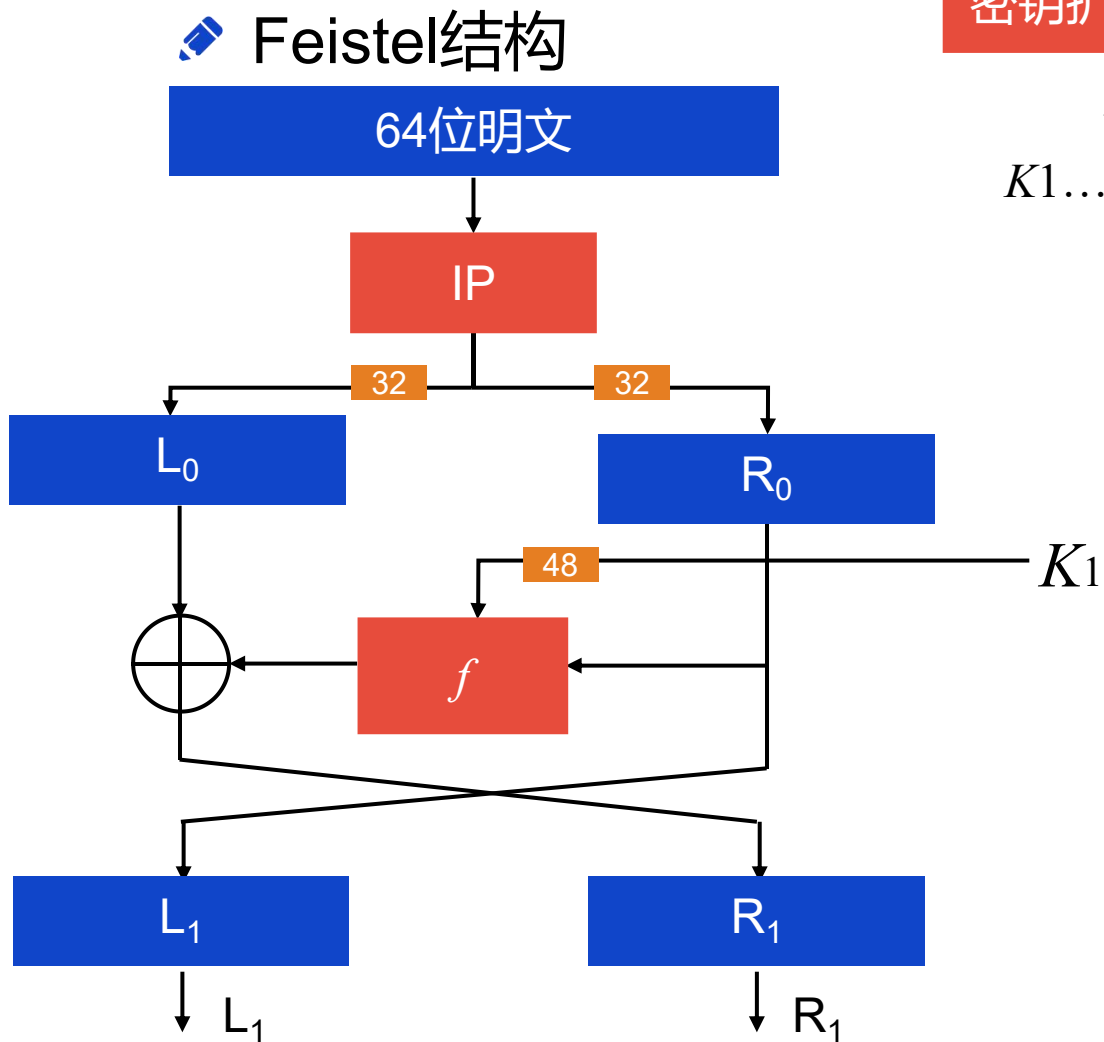
weihu/zhudan@nwpu.edu.cn

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



- ✦ 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

知识回顾 - DES算法结构



-  掌握DES的整体算法结构：分组长度、IP、轮函数、 IP^{-1} 、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

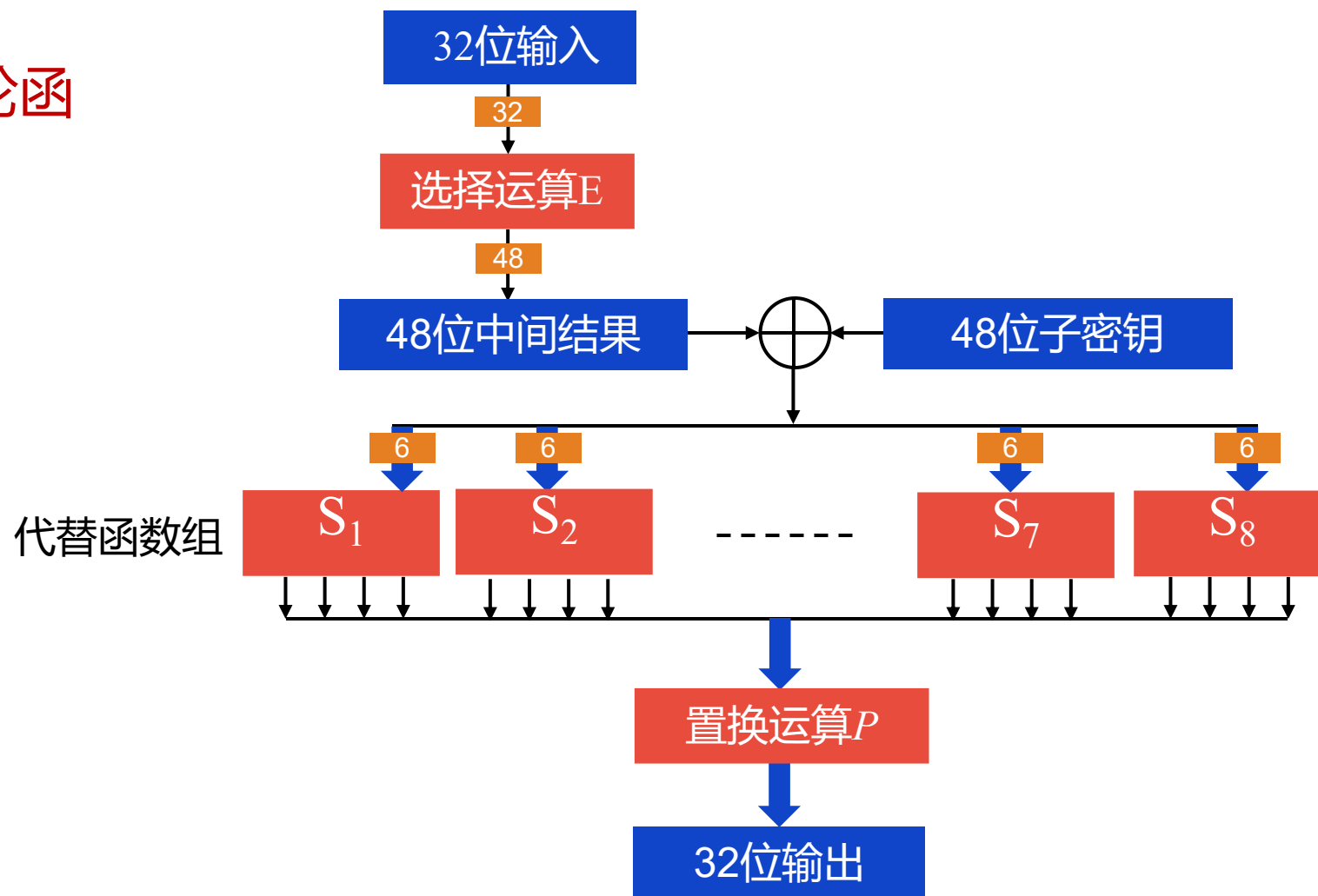
- 逆初始置换IP⁻¹: IP与IP⁻¹互逆
- 例: 在IP中把输入的第1位置换到第40位, 而在IP⁻¹中把输入的第40位置换回第1位
- 保密作用不大: 没有密钥参与, IP和IP⁻¹均公开, 保密意义不大

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

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



- 理解函数对于DES算法安全性的重要作用, 非线性函数
- S盒的混淆作用, S盒和P盒结合使用起到混淆和扩散的效果

- ✎ 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)
 - 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

✎ 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

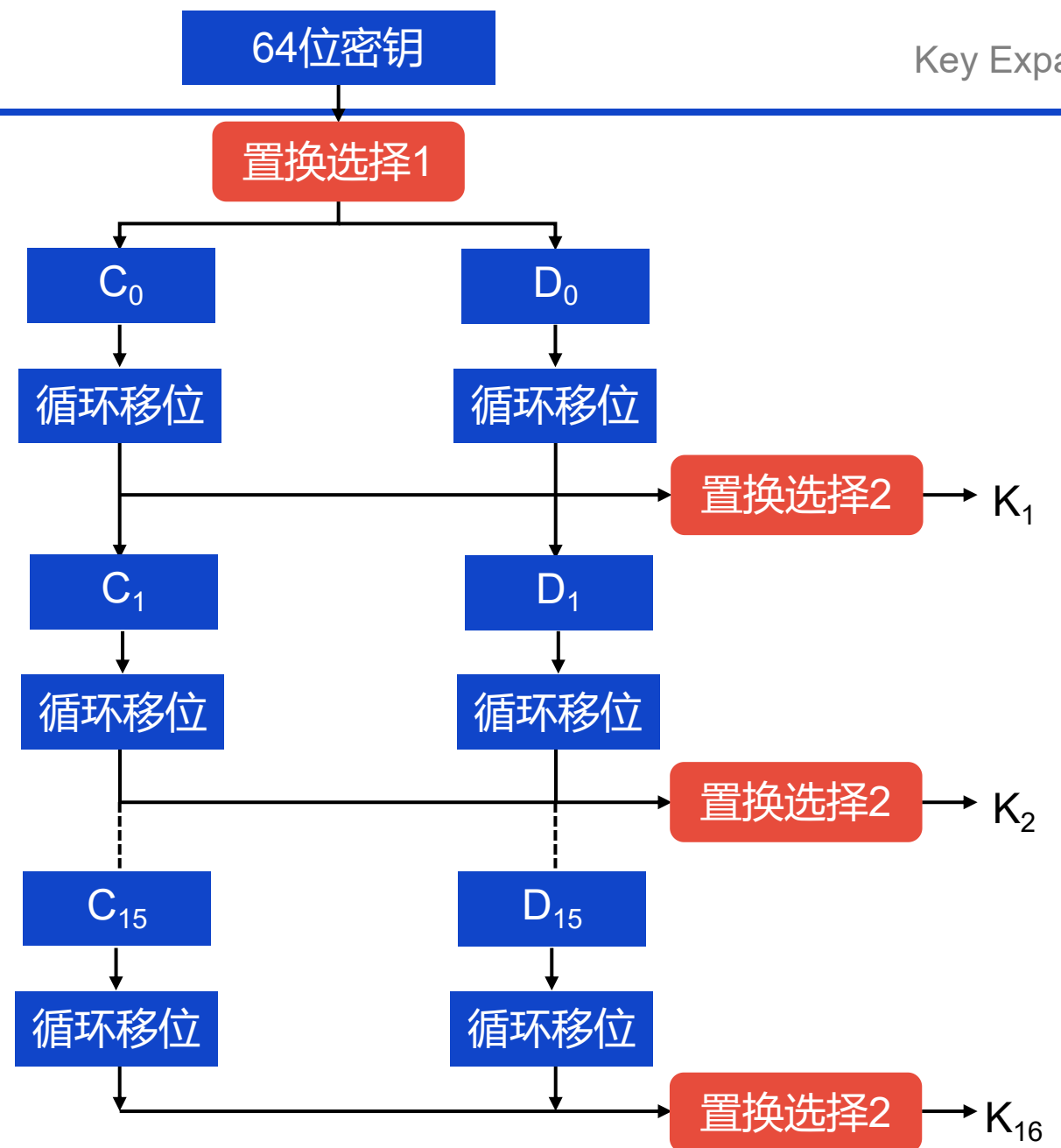
✎ S盒与P置换的互相配合，共同确保DES的安全

✎ 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

- 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}




- ✎ 置换选择1
 - ✎ 去掉密钥中的8个奇偶校验位（有效密钥长度56位）
 - ✎ 打乱重排，形成C₀(左28位)， D₀(右28位)

✎ 置换矩阵

C ₀							D ₀						
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C₀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48位的子密钥 K_i

✎ 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

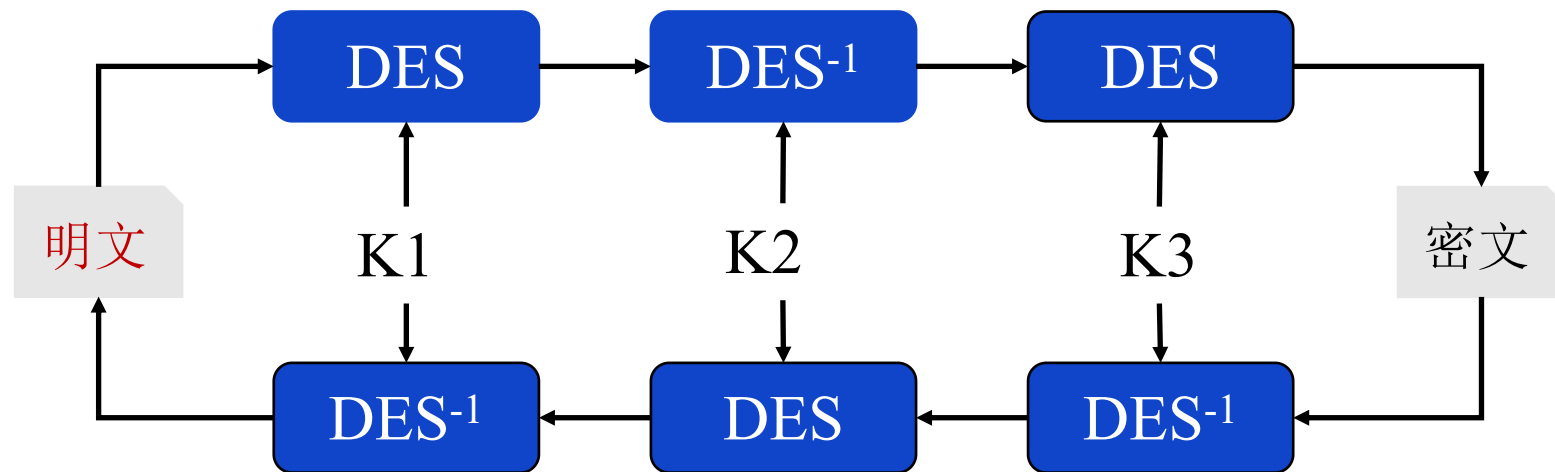
$$\text{设 } C = \text{DES}(M, K), \text{ 则有 } \bar{C} = \text{DES}(\bar{M}, \bar{K})$$

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

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

✎ 了解并理解DES算法存在的安全脆弱性，最大的脆弱点在于密钥太短

- 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属于何种密码网络结构 () ?
- ✎ DES的有效密钥长度为 () ?
- ✎ DES的轮密钥长度为 () ?
- ✎ 3-DES支持的密钥长度为 () ?
- ✎ DES代替函数组S (S盒) 的输入为101010, 其输出为 () ?

	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

章节安排

Outline



AES算法概述



AES数学基础



AES加解密算法



AES密钥扩展算法



AES安全性分析

章节安排

Outline



AES算法概述



AES数学基础



AES加解密算法



AES密钥扩展算法



AES安全性分析

4.7(1) AES的产生背景

- ✎ 1984年12月，美国里根总统下令由国家安全局（NSA）研制新密码标准，以取代DES
- ✎ 1991年新密码开始试用并征求意见
- ✎ 1994年颁布新密码标准（EES, Escrowed Encryption Standard）
 - ✎ 不公开算法，只提供芯片
 - ✎ 每个芯片中的单元（或用户）密钥在政府完全控制之下
 - ✎ 通过法律允许可破译监听
- ✎ 1995年5月，贝尔实验室的博士生M. Blaze在PC 机上用45分钟攻击法律监督字段获得成功
- ✎ 1995年7月美国政府放弃用EES加密数据

- ✦ 1984年12月，里根总统下令由国家安全局（NSA）研制新密码标准，以取代DES
- ✦ 2022年6月，西北工业大学声明遭受来自美国国家安全局（NSA）的网络攻击



- ✎ 1997年美国国家标准与技术研究所（NIST）向社会公开征集高级数据加密标准（AES, Advanced Encryption Standard）
- ✎ 基本要求：比3-DES快，至少和3-DES一样安全，数据分组长度为128位，密钥长度支持128/192/256位
 - ✎ 第一轮：1998年8月20日从应征的21个算法中选出15个
 - ✎ 第二轮：1999年8月又选出其中5个候选算法（RC6, Rijndael, SERPENT, Twofish和MARS）
 - ✎ 第三轮：2000年10月2日再选出1个算法（Rijndael）
- ✎ 2001年11月26日，NIST接受 Rijndael 作为标准
- ✎ 2001年12月4日正式公布为联邦标准：FIPS - 197

4.7(2) AES概况

- 从DES到AES，反映了美国商用密码政策的变化
- 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



Kerckhoffs



DES

公开征集
成功



EES

秘密设计
不成功



AES

更大范围公开征集
成功

应用：

- 许多国际组织都采纳为**加密标准**
- 产品形式：**软件**和**硬件**形式
- 物联网**等新兴领域也见应用

安全性：

- AES仍然是目前主流的**数据加密标准**
- AES主要的安全威胁来源于**侧信道攻击**



- AES安全性受到侧信道攻击威胁，为什么它还是主流的数据加密标准？

- ✎ 分组密码：明文和密文分组长度为128位，密钥长度可为128/192/256位
- ✎ 基本轮函数迭代，轮数可为10/12/14（与密钥长度对应）
- ✎ 整体结构：S-P网络结构
- ✎ 综合运用多种密码技术：置换、替代、代数
- ✎ 不是对合运算：加解密算法存在差异
- ✎ 与DES类似，属于面向二进制的密码：便于计算机实现

章节安排

Outline



AES算法概述



AES数学基础



AES加解密算法



AES密钥扩展算法



AES安全性分析

✎ AES基于有限域 $GF(2^8)$

✎ 有限域 $GF(2^8)$ 上元素的 $GF(2)$ 多项式表示

✎ 字节 $B = b_7b_6b_5b_4b_3b_2b_1b_0$ 可表示成 $GF(2)$ 上的多项式

$$b_7x^7 + b_6x^6 + b_5x^5 + b_4x^4 + b_3x^3 + b_2x^2 + b_1x + b_0$$

✎ 例如, 0x57对应的二进制数为, 01010111

✎ 相应的多项式为 $x^6 + x^4 + x^2 + x + 1$

✎ AES基于有限域 $GF(2^8)$

✎ 有限域 $GF(2^8)$ 上元素的 $GF(2)$ 多项式表示




✎ 字节 $B = b_7b_6b_5b_4b_3b_2b_1b_0$ 可表示成 $GF(2)$ 上的多项式

$$b_7x^7 + b_6x^6 + b_5x^5 + b_4x^4 + b_3x^3 + b_2x^2 + b_1x + b_0$$

✎ 练习: 0xA9对应的二进制数为? 10101001

 相应的多项式为? $x^7 + x^5 + x^3 + 1$

有限域GF(2⁸)上的加法




-  对应多项式系数的模2加 (异或)
-  结果仍为GF(2⁸)上的元素 (次数不超过7的多项式)
-  例, $0x57 + 0x83 = ?$

$$01010111 \oplus 10000011 = 11010100$$

$$(x^6 + x^4 + x^2 + x + 1) \oplus (x^7 + x + 1) = x^7 + x^6 + x^4 + x^2$$

$$0x57 \quad + \quad 0x83 \quad = \quad 0xD4$$

有限域GF(2⁸)上的加法

-  对应多项式系数的模2加 (异或)
-  结果仍为GF(2⁸)上的元素 (次数不超过7的多项式)
-  练习: $0x2A + 0xD7 = ?$

$$00101010 \oplus 11010111 = 11111101$$

$$(x^5 + x^3 + \cancel{x}) \oplus (x^7 + x^6 + x^4 + x^2 + \cancel{x} + 1) = x^7 + x^6 + x^5 + x^4 + x^3 + x^2 + 1$$

$$0x2A + 0xD7 = 0xFD$$

✎ 有限域GF(2⁸)上的乘法

- ✎ 先确定一个8次不可约多项式 $m(x)$
- ✎ AES选择 $m(x) = x^8 + x^4 + x^3 + x + 1$ ，其16进制表示为0x11B
- ✎ 多项式乘法对 $m(x)$ 取模，结果仍为GF(2⁸)上的元素（次数不超过7的多项式）
- ✎ 例， $0x57 \times 0x83 = ?$

$$(x^6 + x^4 + x^2 + x + 1) \otimes (x^7 + x + 1) = x^7 + x^6 + 1 \mod m(x)$$

$$\pencil m(x) = x^8 + x^4 + x^3 + x + 1$$

$$\feather \text{例, } 57 \times 83 = ?$$

$$(x^6 + x^4 + x^2 + x + 1) \otimes (x^7 + x + 1) = x^{13} + x^{11} + x^9 + x^8 + \cancel{x^7} + \cancel{x^7} + x^5 + x^3 + \cancel{x^2} + \cancel{x} + x^6 + x^4 + \cancel{x^2} + \cancel{x} + 1$$

$$x^{13} + x^{11} + x^9 + x^8 + x^5 + x^3 + x^6 + x^4 + 1 - m(x) x^5 = \cancel{x^{13}} + x^{11} + \cancel{x^9} + \cancel{x^8} + \cancel{x^5} + x^3 + \cancel{x^6} + x^4 + 1 + \cancel{x^{13}} + \cancel{x^9} + \cancel{x^8} + \cancel{x^6} + \cancel{x^5} = x^{11} + x^3 + x^4 + 1$$

$$x^{11} + x^3 + x^4 + 1 - m(x) x^3 = \cancel{x^{11}} + \cancel{x^3} + \cancel{x^4} + 1 + \cancel{x^{11}} + x^7 + x^6 + \cancel{x^4} + \cancel{x^3} = x^7 + x^6 + 1$$

$$(x^6 + x^4 + x^2 + x + 1) \otimes (x^7 + x + 1) = x^7 + x^6 + 1 \pmod{m(x)}$$

$$57 \times 83 = C1$$

$$\pencil m(x) = x^8 + x^4 + x^3 + x + 1$$

$\text{例, } 92 \times B4 = ?$

$$(x^7 + x^4 + x) \otimes (x^7 + x^5 + x^4 + x^2) = x^{14} + \cancel{x^{11}} + \cancel{x^8} + x^{12} + \cancel{x^9} + \cancel{x^6} + \cancel{x^{11}} + \cancel{x^8} + x^5 + \cancel{x^9} + \cancel{x^6} + x^3 = x^{14} + x^{12} + x^5 + x^3$$

$$x^{14} + x^{12} + x^5 + x^3 - m(x) x^6 = \cancel{x^{14}} + x^{12} + x^5 + x^3 + \cancel{x^{14}} + x^{10} + x^9 + x^7 + x^6 = x^{12} + x^5 + x^3 + x^{10} + x^9 + x^7 + x^6$$

$$\begin{aligned} x^{12} + x^{10} + x^9 + x^7 + x^6 + x^5 + x^3 - m(x) x^4 &= \cancel{x^{12}} + x^{10} + x^9 + \cancel{x^7} + x^6 + \cancel{x^5} + x^3 + \cancel{x^{12}} + x^8 + \cancel{x^7} + \cancel{x^5} + x^4 \\ &= x^{10} + x^9 + x^6 + x^8 + x^4 + x^3 \end{aligned}$$

$$\begin{aligned} x^{10} + x^9 + x^6 + x^8 + x^4 + x^3 - m(x) x^2 - m(x) &= \cancel{x^{10}} + x^9 + \cancel{x^6} + \cancel{x^8} + \cancel{x^4} + \cancel{x^3} + \cancel{x^{10}} + \cancel{x^6} + x^5 + \cancel{x^3} + x^2 + \\ &\quad \cancel{x^8} + \cancel{x^4} + x^3 + x + 1 = x^9 + x^5 + x^3 + x^2 + x + 1 \end{aligned}$$

$$x^9 + x^5 + x^3 + x^2 + x + 1 - m(x) x = \cancel{x^9} + \cancel{x^5} + x^3 + \cancel{x^2} + \cancel{x} + 1 + \cancel{x^9} + \cancel{x^5} + x^4 + \cancel{x^2} + \cancel{x} = x^4 + x^3 + 1$$

$$(x^7 + x^4 + x) \otimes (x^7 + x^5 + x^4 + x^2) = x^4 + x^3 + 1 \pmod{m(x)} \quad 92 \times B4 = 19$$

\pencil 有限域上的乘法：多项式乘法展开，然后对 $m(x)$ 取模，保证结果的次数不超过7，任然在 $GF(2^8)$ 上

\pencil 加法或者减法均为异或操作

✎ 乘法逆元:

✎ 设 $a(x)$ 的逆元为 $b(x)$, 则 $a(x)b(x) = 1 \bmod m(x)$

✎ 可根据广义**Euclid**算法求出 $b(x)$

✎ 有限域GF(2⁸)上的x乘法(xtime), 定义为

$$\begin{aligned} x \otimes (b_7x^7 + b_6x^6 + b_5x^5 + b_4x^4 + b_3x^3 + b_2x^2 + b_1x + b_0) \\ = b_7x^8 + b_6x^7 + b_5x^6 + b_4x^5 + b_3x^4 + b_2x^3 + b_1x^2 + b_0x \end{aligned}$$

✎ 计算规则:

- ✎ 若 $b_7 = 0$, 次数不超过7, 直接得到结果
- ✎ 否则, 乘法结果减去 $m(x)$, 即与 $m(x)$ 做异或

✎ 通过系数直接计算

- ✎ $B = b_7b_6b_5b_4b_3b_2b_1b_0$ 左移一位, 最低位补0 (乘2)
- ✎ 若 $b_7 = 0$, 直接得到结果
- ✎ 否则, $b_6b_5b_4b_3b_2b_1b_0$ 再与0x1B做异或
- ✎ x 的更高次的乘法可以重复应用xtime实现

xtime(37) = ? 6E

xtime(C5) = ? 91

✎ $xtime$ 计算举例, 计算 57×13

✎ $13 = 01 \oplus 02 \oplus 10$

✎ $57 \times 01 = 57$

✎ $57 \times 02 = xtime(57) = AE$

✎ $57 \times 04 = xtime(AE) = (AE \ll 1) \oplus 1B = 47$

✎ $57 \times 08 = xtime(47) = 8E$

✎ $57 \times 10 = xtime(8E) = 07$

✎ $57 \times 13 = 57 \times (01 \oplus 02 \oplus 10) = 57 \oplus AE \oplus 07 = FE$

✎ $xtime$ 计算练习, 计算 $A2 \times 09$

✎ $09 = 01 \oplus 08$

✎ $A2 \times 01 = A2$

✎ $A2 \times 02 = xtime(A2) = (A2 \ll 1) \oplus 1B = 5F$

✎ $A2 \times 04 = xtime(5F) = (5F \ll 1) = BE$

✎ $A2 \times 08 = xtime(BE) = (BE \ll 1) \oplus 1B = 67$

✎ $A2 \times 09 = A2 \times (01 \oplus 08) = A2 \oplus 67 = C5$

✎ AES数据处理的单位是**字节 (byte)**、**字 (word)** 和**状态 (state)**

- ✎ 一个字 = 4个字节 = 32位，状态为128位
- ✎ 一个字可表示为系数取自GF(2⁸)上的次数低于**4次的多项式**
- ✎ 例，字： 57 83 4A D1 -- $57x^3 + 83x^2 + 4Ax + D1$

✎ 字加法：**两多项式系数按位模2加**

- ✎ 例， $(57x^3 + 83x^2 + 4Ax + D1) + (Ax^3 + B3x^2 + EF)$
 $= 5Dx^3 + 30x^2 + 4Ax + 3E$

✎ 状态 (128位)

- ✎ 加解密过程中的中间数据
- ✎ 以字节为元素的矩阵或二维数组

状态矩阵

a_{0,0}	a_{0,1}	a_{0,2}	a_{0,3}
a_{1,0}	a_{1,1}	a_{1,2}	a_{1,3}
a_{2,0}	a_{2,1}	a_{2,2}	a_{2,3}
a_{3,0}	a_{3,1}	a_{3,2}	a_{3,3}

✎ 字乘法：设 a 和 c 是两个字， $a(x)$ 和 $c(x)$ 为对应的字多项式，AES定义 a 和 c 的乘积 b 为

$$b(x) = a(x)c(x) \bmod x^4 + 1$$

✎ 假设

$$a(x) = a_3x^3 + a_2x^2 + a_1x + a_0$$

$$c(x) = c_3x^3 + c_2x^2 + c_1x + c_0$$

$$b(x) = b_3x^3 + b_2x^2 + b_1x + b_0$$

✎ 则， $b(x) = a(x)c(x) \bmod x^4 + 1$ 为

$$b_0 = a_0c_0 + a_3c_1 + a_2c_2 + a_1c_3$$

四次项和常量

$$b_1 = a_1c_0 + a_0c_1 + a_3c_2 + a_2c_3$$

一次项

$$b_2 = a_2c_0 + a_1c_1 + a_0c_2 + a_3c_3$$

二次项

$$b_3 = a_3c_0 + a_2c_1 + a_1c_2 + a_0c_3$$

三次项

✎ 模多项式是4次多项式 $x^4 + 1$ ，因此，相乘结果为4次的项取模约简为常量

✎ $b_0 \sim b_3$ 分别对应取模约简后的常量至3次项

✎ 字乘法的矩阵表示

✎ $x^4 + 1 = (x^2 + 1)(x^2 + 1)$, 是可约多项式, 字 $c(x)$ 不一定存在逆元

✎ $c(x)$ 存在逆元的条件是 $(x^4+1, c(x)) = 1$

✎ AES选择的 $c(x)$ 有逆, $c(x) = 03x^3 + 01x^2 + 01x + 02$

$$\begin{pmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{pmatrix} = \begin{pmatrix} c_0 & c_3 & c_2 & c_1 \\ c_1 & c_0 & c_3 & c_2 \\ c_2 & c_1 & c_0 & c_3 \\ c_3 & c_2 & c_1 & c_0 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{pmatrix}$$

✎ 字的 x 乘法：设 $b(x)$ 是一个字，

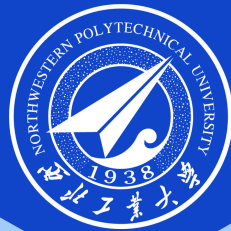
$$\begin{aligned} p(x) &= xb(x) \bmod x^4 + 1 \\ &= b_3x^4 + b_2x^3 + b_1x^2 + b_0x \bmod x^4 + 1 \\ &= b_2x^3 + b_1x^2 + b_0x + b_3 \end{aligned}$$

✎ 因为模 $x^4 + 1$ ，字的 x 乘法相当于按字节循环移位

✎ 写成矩阵形式

$$\begin{pmatrix} p_0 \\ p_1 \\ p_2 \\ p_3 \end{pmatrix} = \begin{pmatrix} 00 & 00 & 00 & 01 \\ 01 & 00 & 00 & 00 \\ 00 & 01 & 00 & 00 \\ 00 & 00 & 01 & 00 \end{pmatrix} \begin{pmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{pmatrix}$$

- ✎ FIPS 197 Advance Encryption Standard (AES),
https://www.nist.org/nist_plugins/content/content.php?content.39
- ✎ Mini-AES,
https://doc.sagemath.org/html/en/reference/cryptography/sage/crypto/block_cipher/miniaes.html
- ✎ R. C.-W. Phan. Mini advanced encryption standard (mini-AES): a testbed for cryptanalysis students. *Cryptologia*, 26(4):283–306, 2002



感谢聆听!

THANK YOU FOR YOUR ATTENTION!