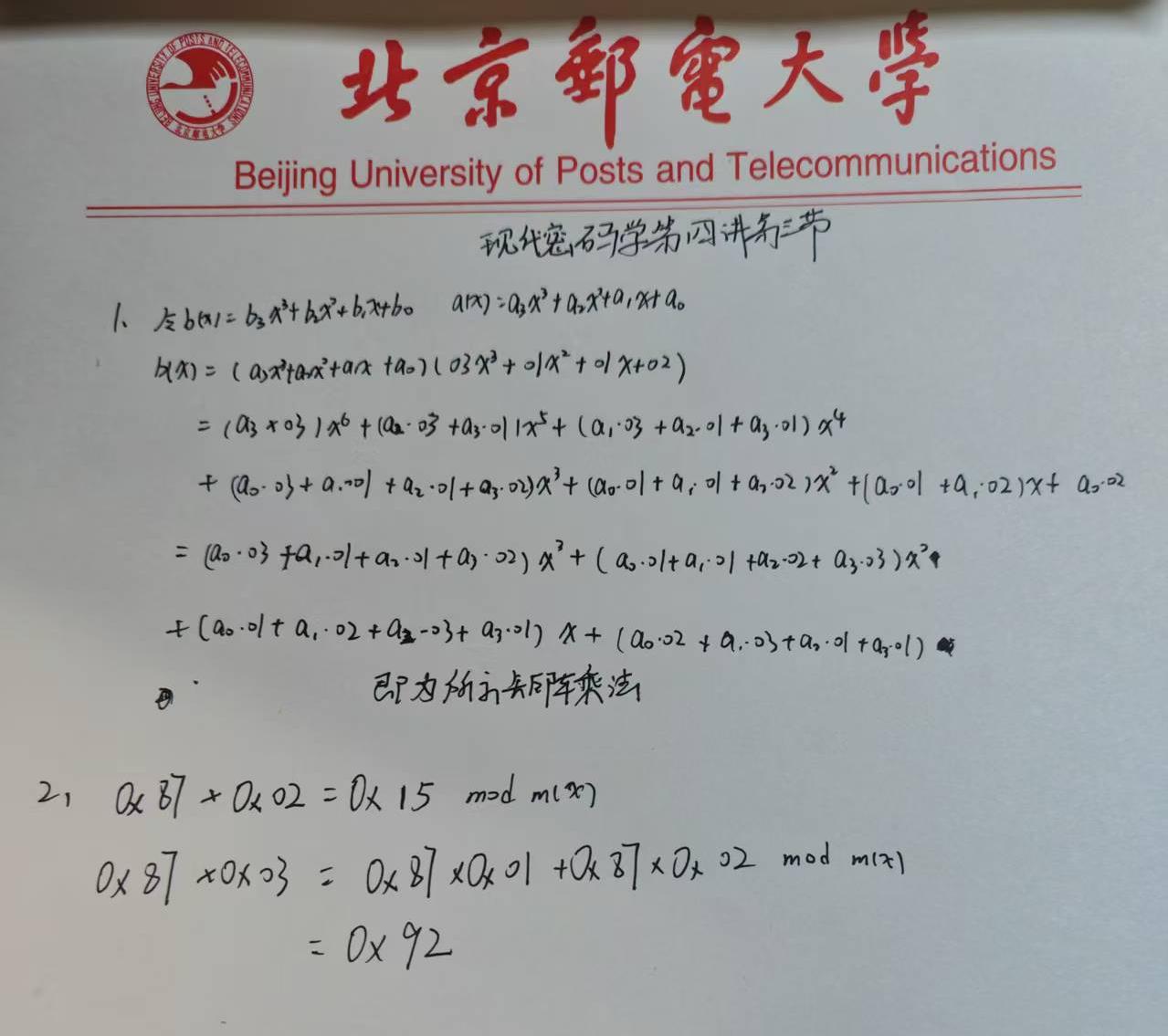
**第三节：**

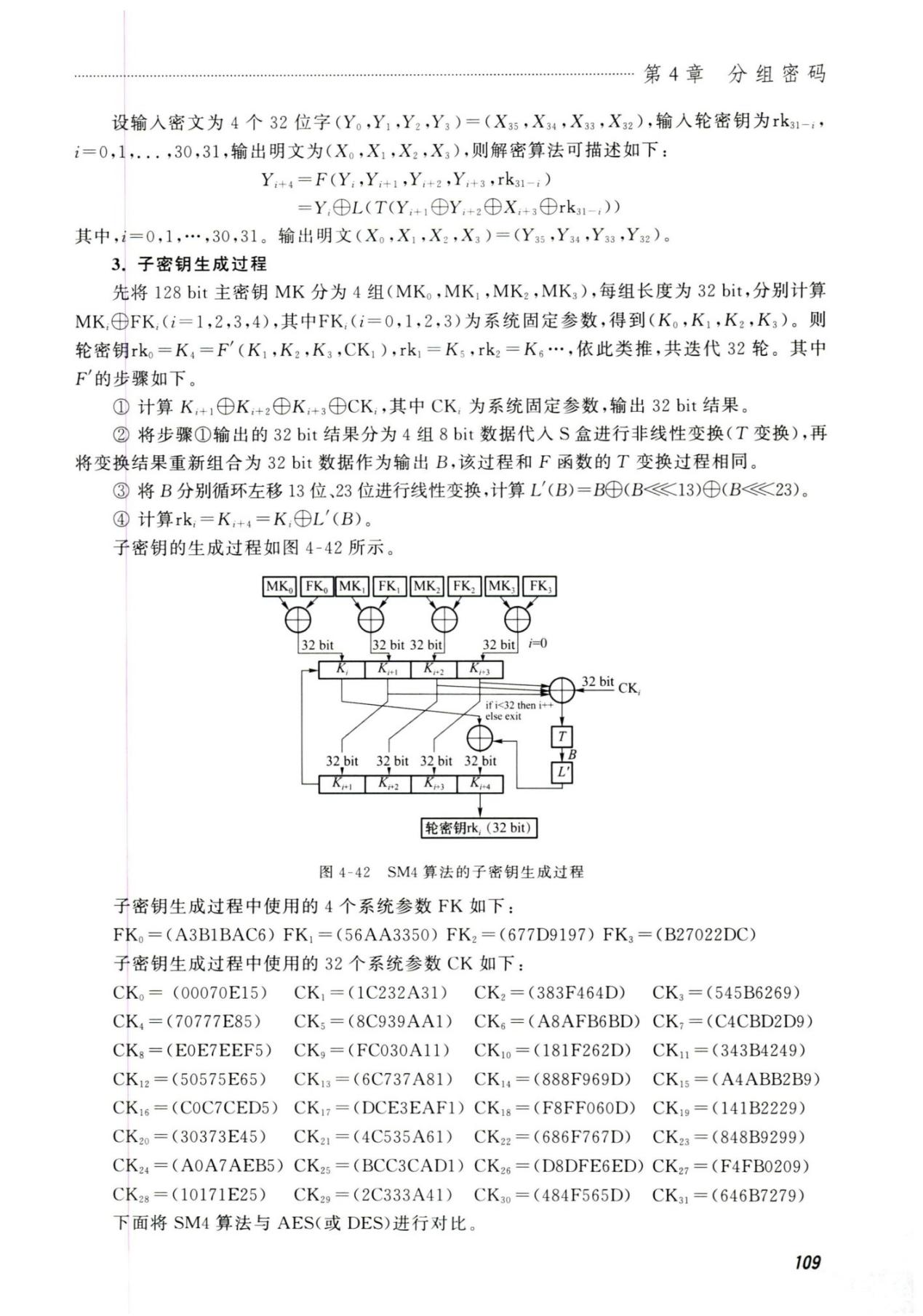
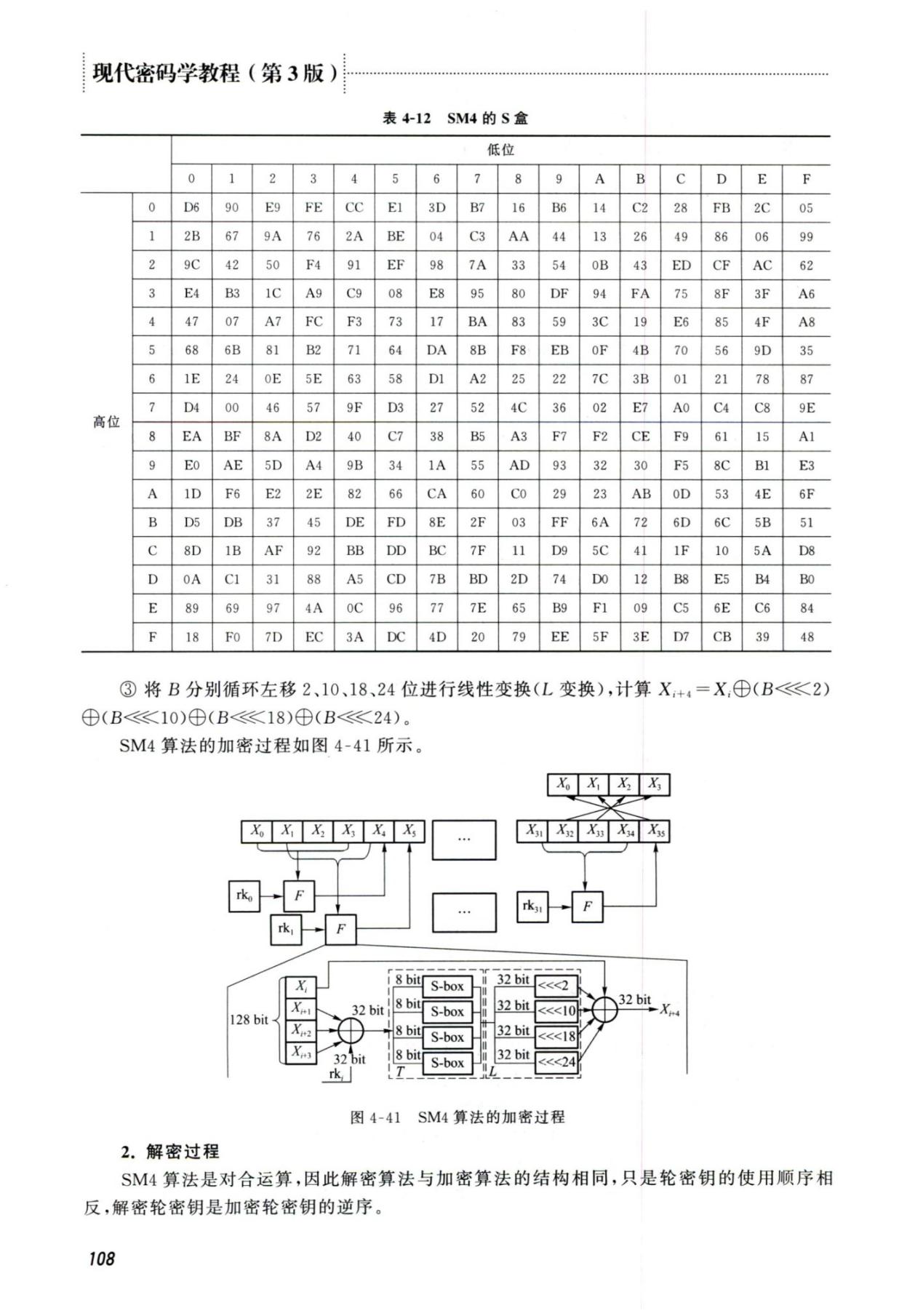
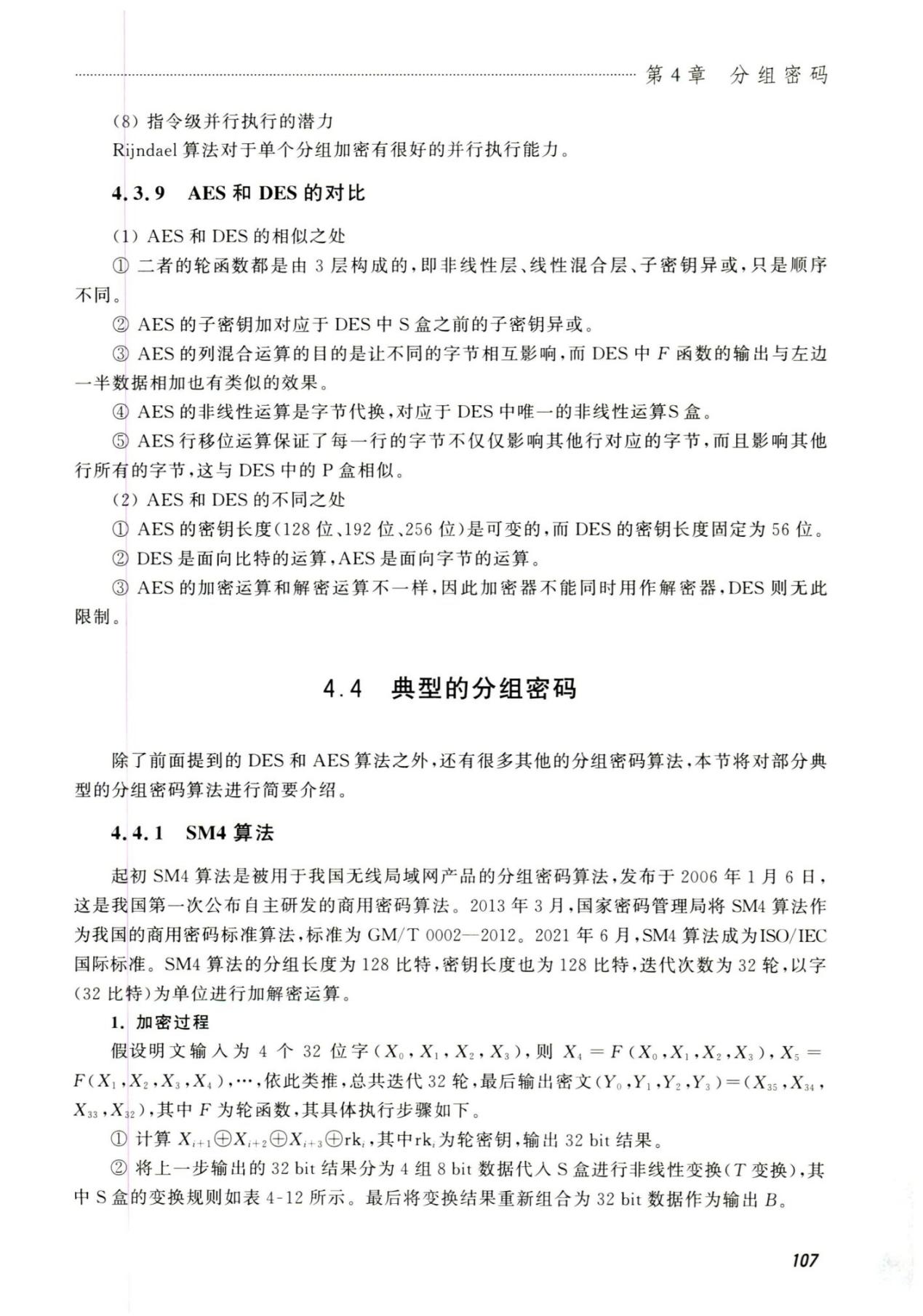
****

**3.调研SM4算法，其迭代结构属于何类型？并详细描述加解密及密钥编排的步骤。**

1.迭代结构：

SM4采用32轮Feistel结构，每轮使用一个子密钥进行非线性变换和线性变换。

2.加解密过程及密钥编排过程：



**4.使用乘法逆元及仿射方法实现AES字节代换操作的快速运算方法。**

def gf\_mul(a, b):

"""

在 GF(2^8) 中计算 a 和 b 的乘积，

使用不可约多项式 m(x)=x^8+x^4+x^3+x+1（0x11B），

当 a 高位为 1 时，使用 0x1B 进行约简。

"""

p = 0

for \_ in range(8):

if b & 1:

p ^= a

hi\_bit\_set = a & 0x80

a = (a << 1) & 0xFF

if hi\_bit\_set:

a ^= 0x1B

b >>= 1

return p

def gf\_inv(a):

"""

计算 GF(2^8) 中 a 的乘法逆元，

对于 a=0 定义逆元为 0。

采用暴力搜索的方法找到满足 gf\_mul(a, inv) == 1 的 inv。

"""

if a == 0:

return 0

for inv in range(1, 256):

if gf\_mul(a, inv) == 1:

return inv

return 0 # 理论上不会到达此处

def rotl(x, shift):

"""

实现 8 位数 x 的循环左移操作。

"""

return ((x << shift) & 0xFF) | (x >> (8 - shift))

def aes\_sbox(x):

"""

对输入字节 x 先计算其在 GF(2^8) 中的乘法逆元，

再进行仿射变换（多次循环左移异或，并最终与 0x63 异或）。

"""

inv = gf\_inv(x)

# 仿射变换：inv 与其 1,2,3,4 位循环左移的结果异或，再异或常数 0x63

y = inv ^ rotl(inv, 1) ^ rotl(inv, 2) ^ rotl(inv, 3) ^ rotl(inv, 4) ^ 0x63

return y & 0xFF

def main():

s = input("请输入一个字节（十六进制，例如 0x57）：")

try:

byte = int(s, 16)

except ValueError:

print("输入错误，请输入有效的十六进制数。")

return

sbox = aes\_sbox(byte)

print("字节代换结果为：0x{:02X}".format(sbox))

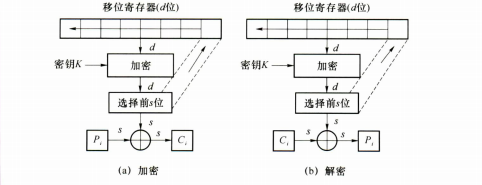
if \_\_name\_\_ == '\_\_main\_\_':

main()

**第四节：**

**1.调研OFB模式并分析其各自性质。**

输出反馈模式是基于分组密码的同步序列密码算法的一种例子。它与CFB模式相似,但是将前一次加密产生的s比特输出分组中的非密文分组送入移位寄存器最右边的位置。因此,这种方法有时也叫作内部反馈,因为反馈机制独立于明文和密文而存在。



性质：

1.消息作为比特流进行加密，无须分组填充；

2.加密和解密过程只调用加密算法（用于生成密钥流，无需解密算法）；

3.存在密文扩展（仅IV的传输带来额外开销）；

4.密钥流可在接收消息前进行计算，无需按顺序解密；

5.不存在错误传播；

6.发送者和接受者必须保持同步；

7.适合大于一个分组长度的长数据加密，常用于流式数据加密。

**2.调研CCM模式，并比较与课堂所讲工作模式异同。**

CCM（Counter with CBC-MAC）模式是一种认证加密模式，结合了计数器模式（CTR）和CBC-MAC，用于提供机密性和完整性的保护。它常用于需要高安全性和认证的应用，如Wi-Fi 加密（WPA2）。

共同点：

1.分组加密模式：都基于块密码（如 AES）。

2.对称加密：都使用相同的密钥进行加密和解密。

3.适用于数据加密：它们都可以用于加密固定长度的明文块。

不同点：

