

专业特色选修课《网络信息安全》



密码算法和密码学

上篇

En/Decryption Algorithms and Cryptology

嵩 天

songtian@bit.edu.cn

北京理工大学计算机学院

本节大纲

- 密码学的基本概念
- 对称密钥密码算法

密码学的历史

- 密码学是一门历史悠久的学科

Cryptology , Cryptography , from Greek

- 密码技术是研究数据加密、解密及变换的科学，涉及数学、计算机科学、电子与通信等学科
- 加密是把数据和信息转换为不可识别的密文的过程
- 解密是恢复数据和信息本来面目的过程

密码学的历史

- 密码学早在公元前400多年就已经产生
- 希腊人把奴隶的头发剃光，把秘密刺在头上，等头发长起来了再派他上路
- 斯巴达人采用羊皮条和木棍进行加密
- 周朝《六韬》，兵符，藏头诗
- 中国古代的“江湖黑话”



挂彩、溜狗、豆儿、总瓢把子、流月汪则中

密码学的历史

- 古罗马：凯撒密码（Caesar 密码）

字母表

ABCDEFGHIJKLMNOPQRSTUVWXYZ

对应密文

DEFGHIGKLMNOPQRSTUVWXYZABC

明文

Caesar was a great soldier

密文

Fdhvdu zdv d juhdu vroglhu

CAESAR 密码： $c = (m + 3) \text{ Mod } 26$

密码学的历史

- 美国南北战争时期

输入方向 →

输出方向 ↓

C	A	N	Y
O	U	U	N
D	E	R	S
T	A	N	D

明文:

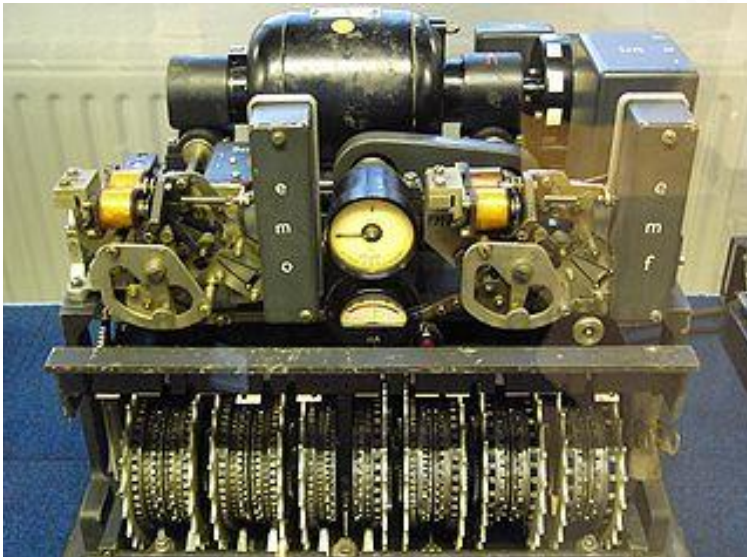
Can you understand

密文:

Codtaueanurnynsd

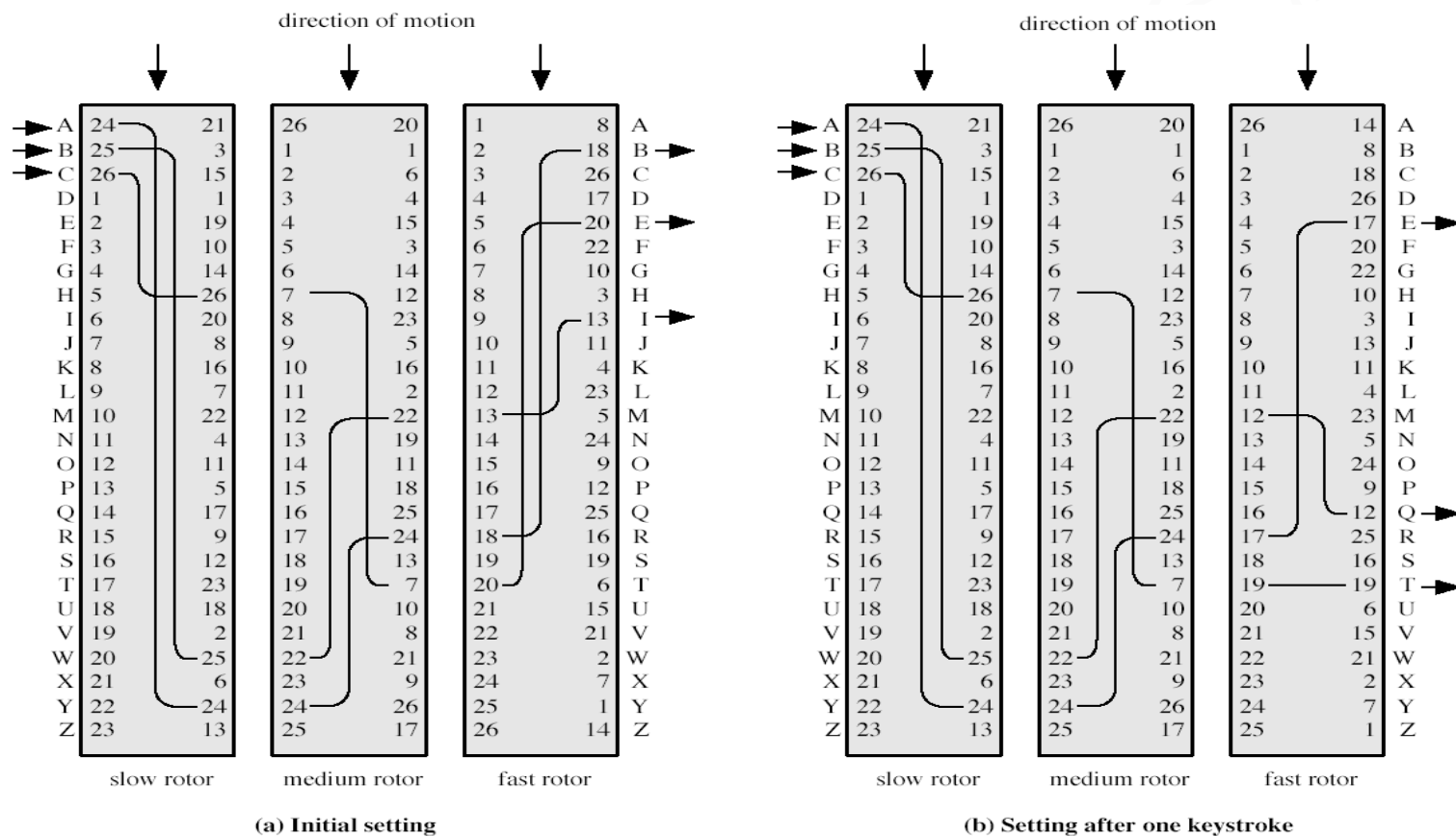
密码学的历史

- 第二次世界大战，德国
 - 恩尼格玛（Enigma）密码机，“谜”
 - 由Arthur Scherbius于1919年发明，1925年生产3万台



密码学的历史

- 具有连线的三转子机器



密码学的历史



- 艾伦·图灵 (Alan Turing)

- 二战期间成功地破译了纳粹德国密码系统Enigma。
- 1912~1954，英国数学家，一生对智能与机器之间的关系进行着不懈探索，被誉为“**计算机科学之父**”。
- 1931~1934，剑桥大学国王学院；1937~1938，普林斯顿大学
- 1936年，24岁的图灵提出“图灵机”的设想。
- 《机器会思考吗？》的论文提出了用于判定机器是否具有智能的试验方法，即**图灵试验**，被誉为“**人工智能之父**”

密码学的历史



- 艾伦·图灵 (Alan Turing)
 - 1948年，奥运会马拉松项目银牌得主。
 - 1954年6月8日，服毒自杀，42岁
 - 1966年，美国计算机协会以他的名字命名了计算机领域的最高奖“图灵奖”。
 - 2000年，姚期智 Andrew Chi-Chih Yao

密码学的历史

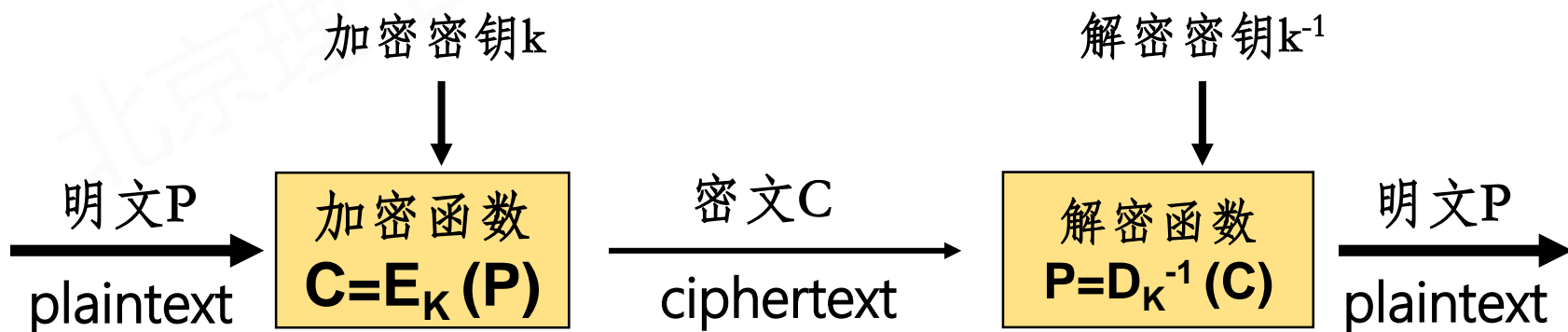
- 现代密码学

- 随着电子计算机的诞生，以语言学为基础的密码方法失效
- 现代密码方法主要在二进制字符串上进行
- 密码方法所基于的问题只要被证明无法有效解出，就是安全
- 现代密码学的主要威胁：
 - 暴力破解的速度
 - 量子计算的发展

密码学的基本概念

- 基本术语

- 明文: plaintext, 原有的信息
- 密文: ciphertext, 明文经过加密变换后的内容
- 加密函数: Encryption, 用来加密的数学函数
- 解密函数: Decryption, 用来解密的数学函数



密码算法的分类

- 古典密码算法和现代密码算法
 - 根据算法和密钥是否分开来区分
- 古典密码算法
 - 密码体制的安全性依赖于算法本身的保密性
 - 算法存在以下限制
 - 不适合大规模生产
 - 不适合较大的或者人员变动频繁的组织
 - 用户无法了解算法的安全性

密码算法的分类

- 古典密码算法种类

- 代码加密：“代码”指专有词汇或者特殊用语，“黑话”
- 替换加密：用一组密文字母代替一组明文字母，但保持明文的顺序，例如：凯撒密码
- 变位加密：对明文字母作重新排序
- 一次性加密：利用代码本，结合上述方法，用代码本上每一页加密明文的一个片段

密码算法的分类

- 现代密码算法

- 把算法和密钥分开
- 密码算法可以公开，密钥保密
- 密码系统的安全性依赖于密钥的保密性
- 优点包括：
 - 密码算法可以经过充分论证，安全性有保证
 - 对密码系统的管理简单
 - 可应用于大规模场景

密码算法的分类

- 对称密钥密码和非对称密钥密码

- 根据加密和解密是否使用相同的密钥来区分

- 对称密钥密码

- 也成“单钥体制”， $k = k^{-1}$

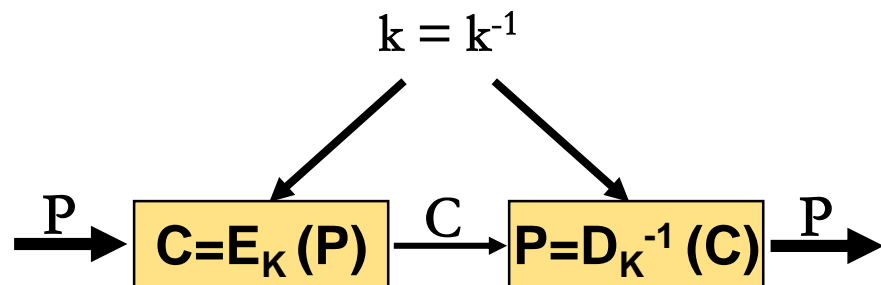
- 算法优点

- 加密速度快，使用简单

- 算法缺点

- 密钥分配：必须通过保密通道进行

- 密钥个数： n 个用户需要 $n(n-1)$ 个密钥



密码算法的分类

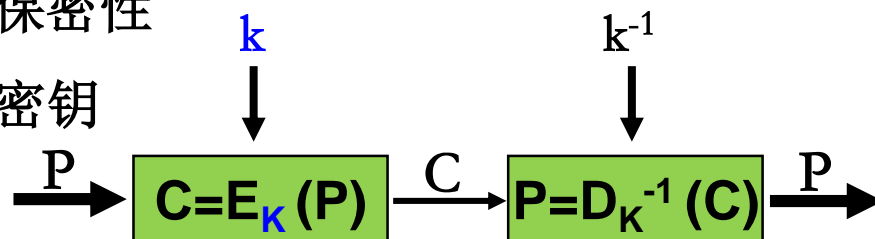
• 非对称密钥密码

- “公开密钥密码体制 (PKI-Public Key Infrastructure)”
- 加密和解密使用不同的密钥 (k, k^{-1})
- 把加密密钥公开 (公钥), 解密密钥保密 (私钥)
- 对于每一个用户, 形成“**公私密钥对**”
- 算法优点

- 密钥分配: 不必保持信道的保密性
- 密钥个数: n 个用户需要 n 个密钥

- 算法缺点

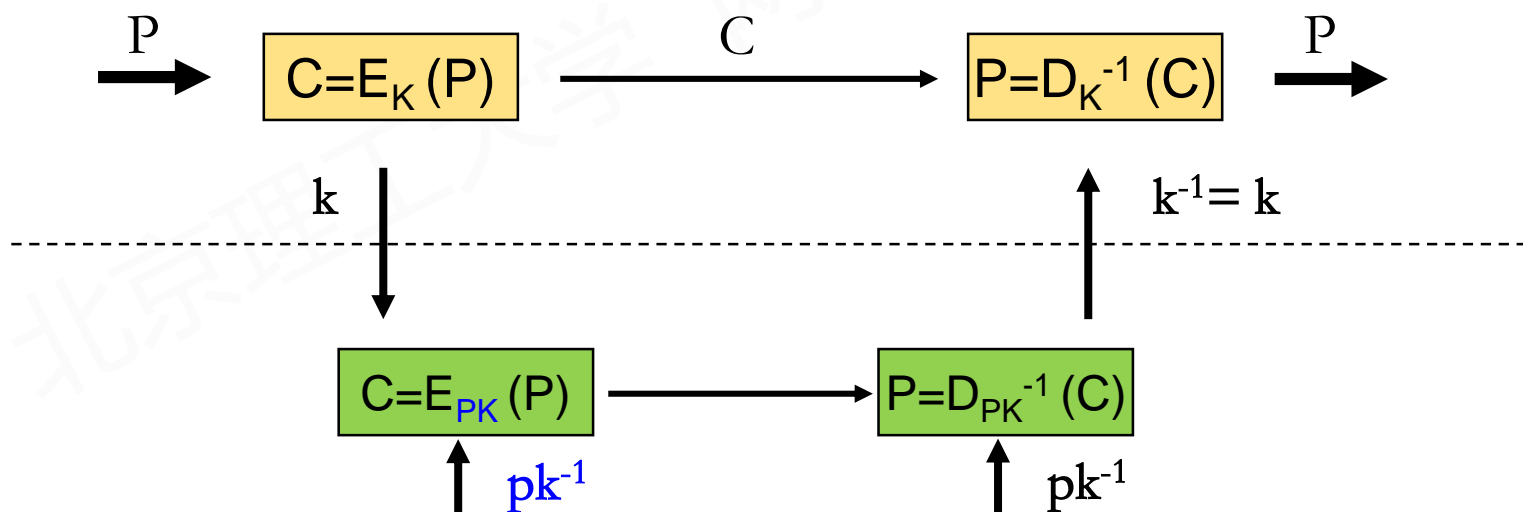
- 算法速度: 比对称密钥算法慢10倍左右(软件)



密码算法的分类

- 混合加密体系

- 结合对称密钥密码和非对称密钥密码体系
- 采用对称方法进行通信，采用PKI体系进行密钥交换



密码算法的分类

- 分组密码和序列密码
 - 根据每次操作的数据单元是否分块来区分
- 分组密码 (Block Cipher)
 - 一次加密或解密操作作用于一个数据块，比如64bit
 - 数据块之间加密是独立的
- 序列密码 (Stream Cipher)
 - 一次加密或解密操作作用于一位或者一个字节
 - 前部分密文参与后部分明文的加密，数据块之间加密不独立

密码算法的分类

- 双向加密和单向加密
 - 根据明文加密后是否需要还原来区分
- 双向加密
 - 明文加密后，需要解密，目的是信息的交换
- 单向加密
 - 明文加密后，不需要解密，目的验证数据的完整性

密码算法的分类

- 古典密码算法和现代密码算法
 - 根据算法和密钥是否分开来区分
- 对称密钥密码和非对称密钥密码
 - 根据加密和解密是否使用相同的密钥来区分
- 分组密码和序列密码
 - 根据每次操作的数据单元是否分块来区分
- 双向加密和单向加密
 - 根据明文加密后是否需要还原来区分

密码分析

- 什么是密码分析？

- 未知密钥，利用数学方法恢复明文，或者推导出密钥
- 对密码进行分析的尝试称为“攻击”或者“破解”
- 对密码体系的攻击方法包括：
 - 暴力破解：对密钥进行穷尽搜索
 - 已知明文攻击：利用一段已知明文和密文的对应关系
 - 选定明文攻击：设法让对手加密一段选定的明文，获得加密结果
 - 选定密文攻击：设法让对手回复一段的密文，获得明文结果
 - 选定密钥攻击：用于分析密码算法和体系

密码分析

- 其他密码攻击方法

- 可以针对人机系统的弱点进行攻击，而不是攻击密码算法本身
- 欺骗用户密码（技术手段或者社会工程学）
- 在用户输入密钥时，窥视或者偷窃密钥内容
- 利用密码系统实现中的缺陷或者漏洞
- 妨碍用户正确使用密码系统
- 让口令的另一方透露密钥和信息
- 威胁用户交出密钥

密码分析

- 密码算法的安全性

- 如果破解算法的代价大于被加密数据本身的价值，或者在信息的生命周期内无法破解，那么算法可能是安全的
- 如果一个密码算法用可得到的资源不能被破解，则称该算法是计算上安全的

- 处理复杂性：计算量、CPU时间
- 数据复杂性：所需输入的数据量
- 存储复杂性：计算所需要的存储空间



本节大纲

- 密码学的基本概念
- 对称密钥密码算法

对称密钥密码算法

- 常用的对称密钥密码算法

- DES、3DES

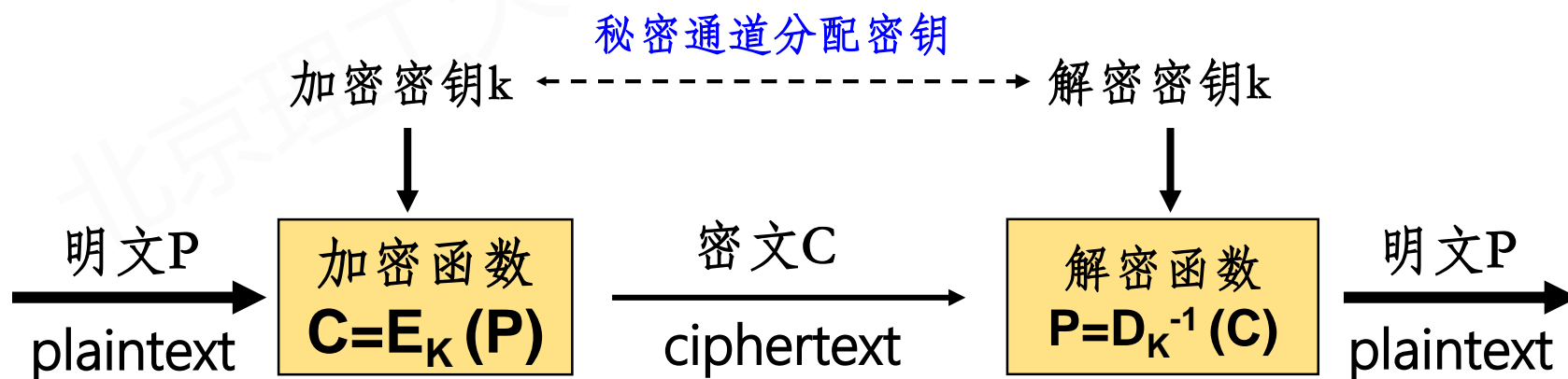
- AES

- IDEA

- CAST-128

- RC2、RC4、RC5

- Blowfish



DES算法

- DES, Data Encryption Standard
- 20世纪70年代, IBM公司为美国国家标准局研制
- 1977年成为美国国家标准, 1998年废弃
- 2001年, DES算法被安全性更强的AES算法所取代
- DES算法是一个分组加密算法, 以64bit为一个分组
- DES算法使用标准的算术和逻辑运算

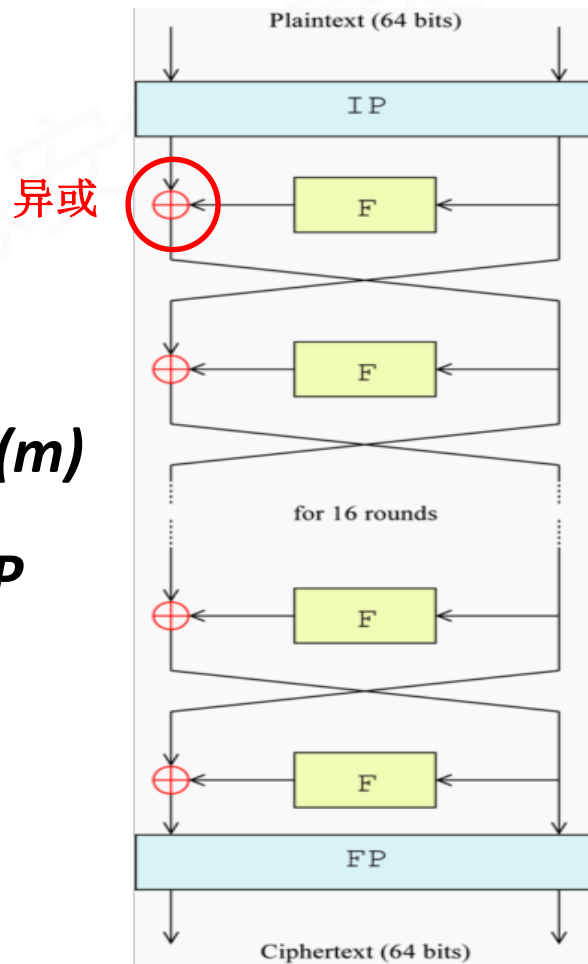
DES算法

- DES的总体框架

- 明文分成64bit为单位的块m
- 对于每个m，执行如下操作：

$$DES(m) = FP \cdot T_{16} \cdot T_{15} \cdot \dots \cdot T_2 \cdot T_1 \cdot IP(m)$$

- 初始置换（Initial Permutation）， IP
- 16轮迭代， T_i ， $i=1,2,\dots,16$
- 末置换（Final Permutation）， FP



DES算法

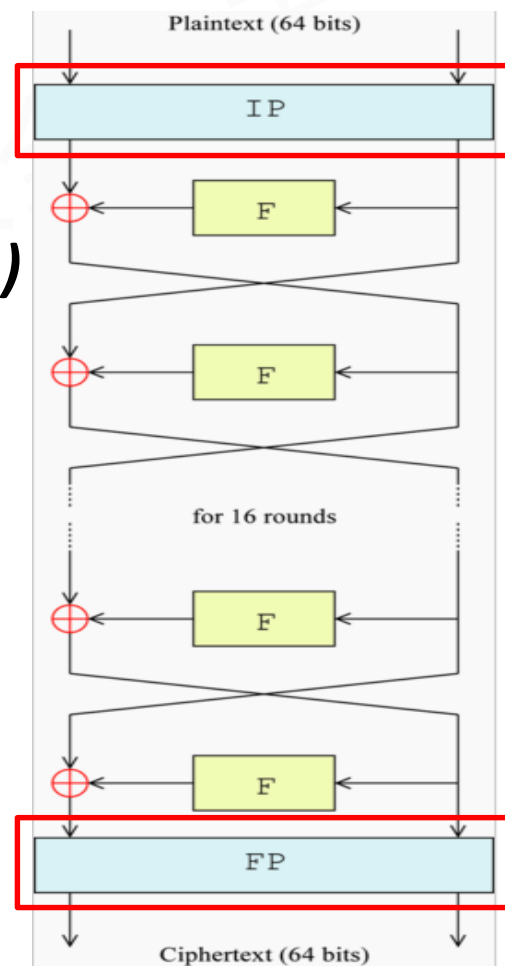
- DES的初始置换和末置换

$$DES(m) = FP \bullet T_{16} \bullet T_{15} \bullet \dots \bullet T_2 \bullet T_1 \bullet IP(m)$$

– $IP \bullet FP = I$, 即 $FP = IP^{-1}$

– 对 m 中64个位置进行置换

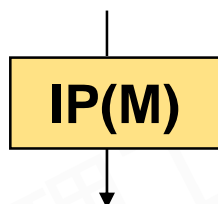
– 输入: $m = m_1 m_2 \dots m_{64}$



DES算法

- DES的初始置换和末置换

$m = m_1 m_2, \dots, m_{62} m_{63}, m_{64}$

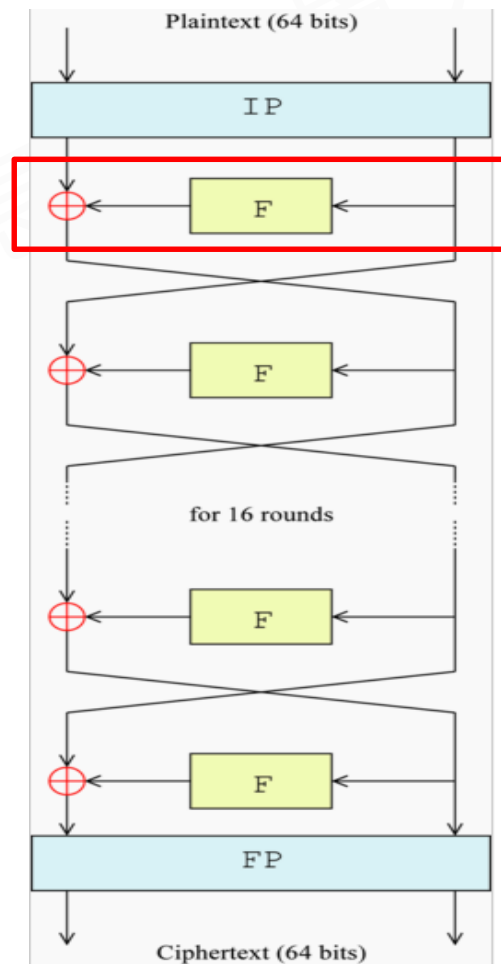
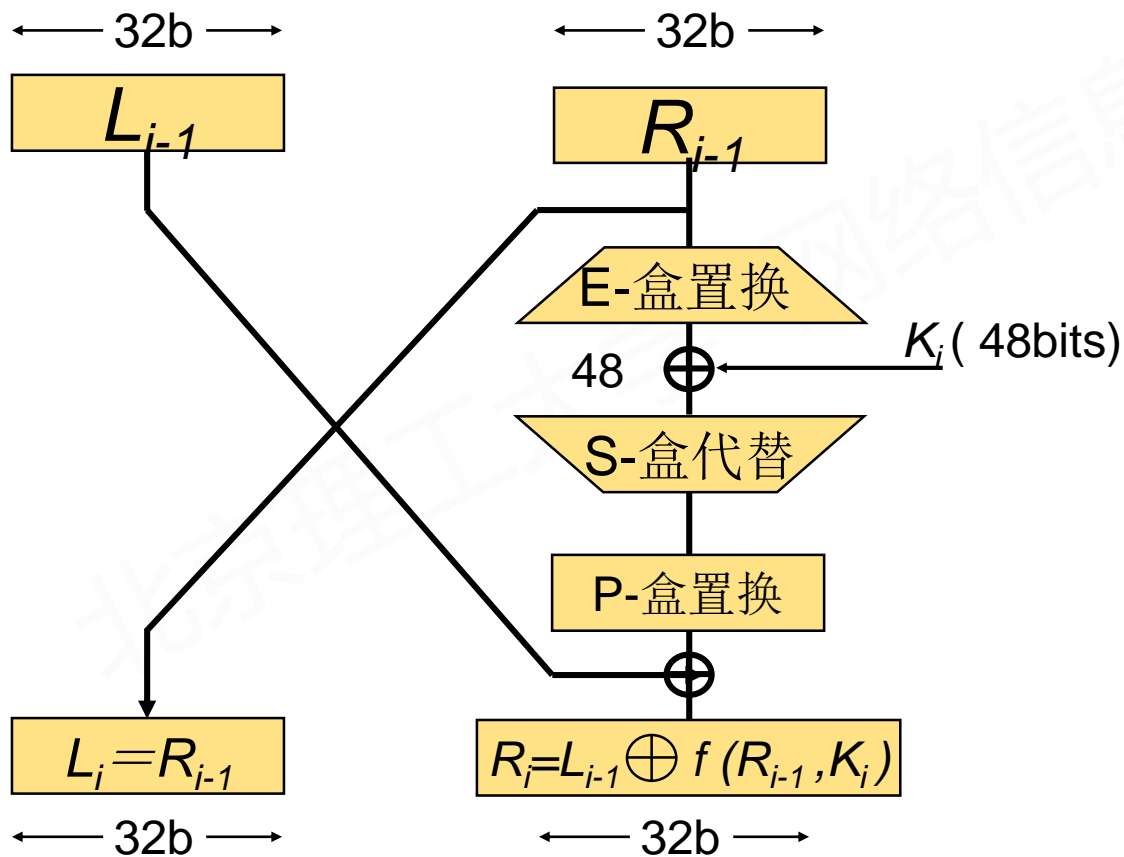


$m' = m_{58} m_{50}, \dots, m_{23} m_{15}, m_7$

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

DES算法

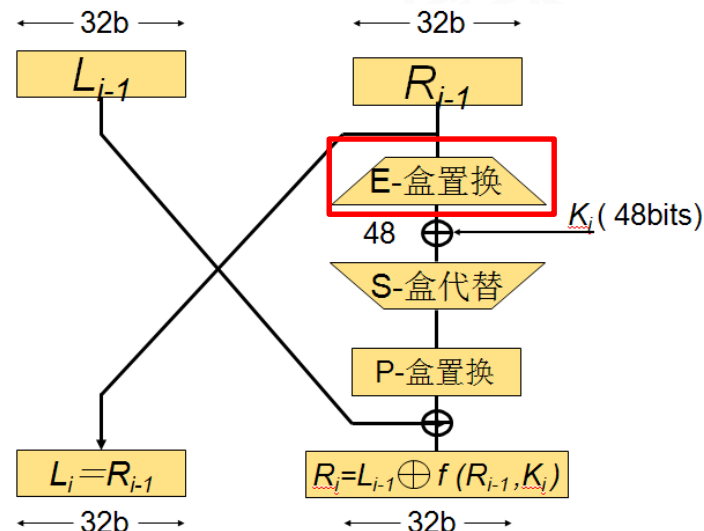
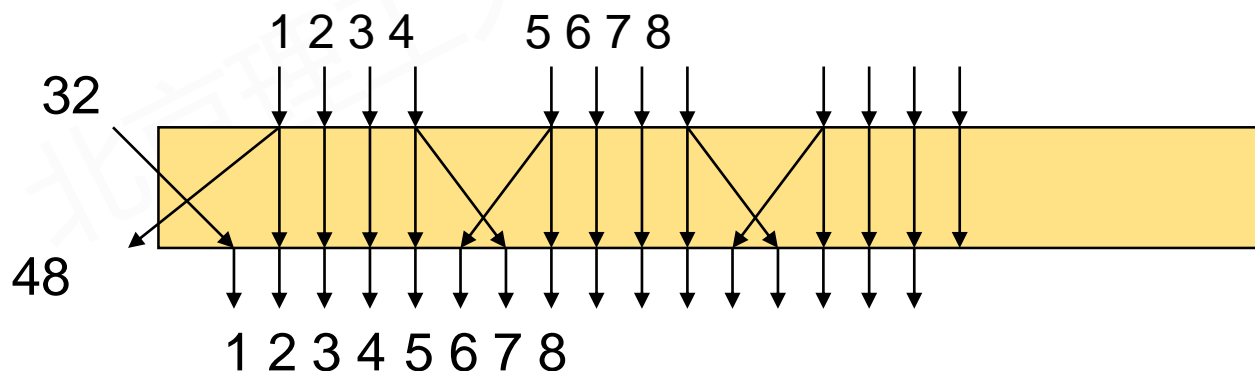
- 每次迭代



DES算法

• E-盒置换

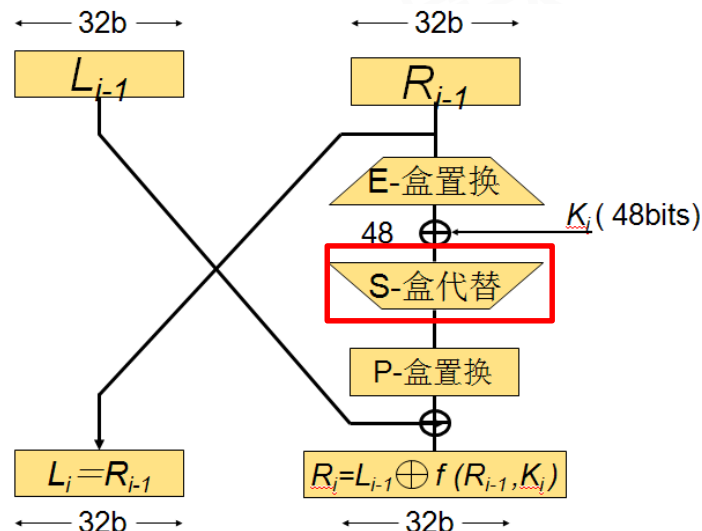
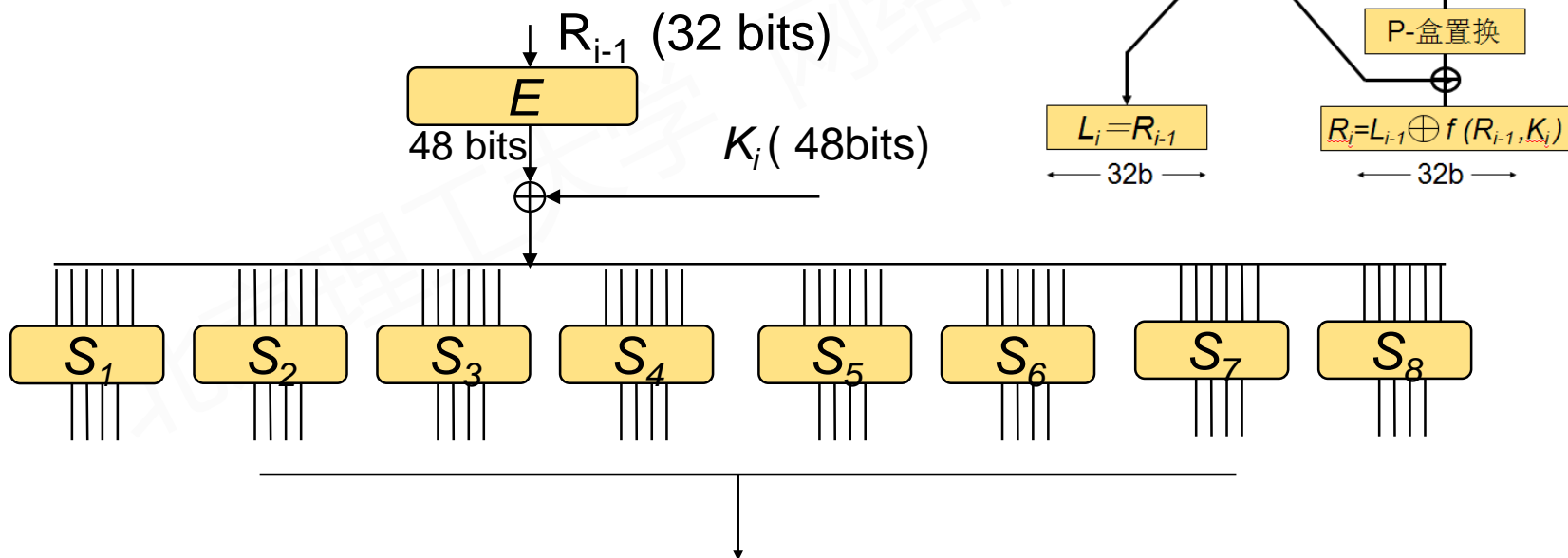
- 将32位扩展成48位
- 目的：输入的一位影响下一步的两个替换，使得输出对输入的依赖性传播得更快，密文的每一位都依赖于明文的每一位



DES算法

- S-盒置换

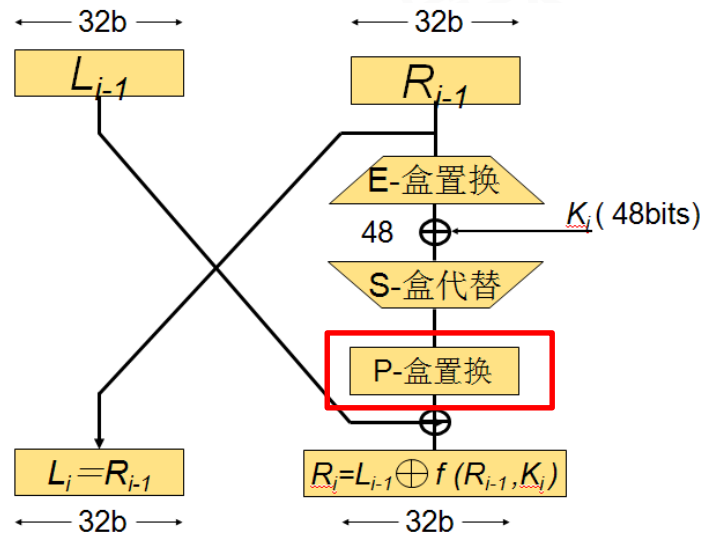
- 将48位压缩成32位
- S_i 输入6位，查表输出4位



DES算法

- P-盒置换

- 32位输入32位输出
- 仅置换位置

[illegible]

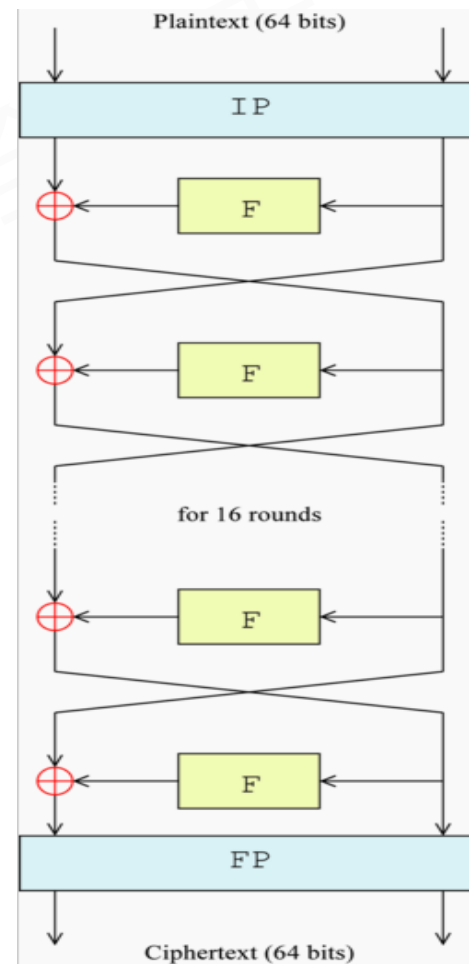
DES算法

- DES的解密过程

$$m = DES^{-1}(c) = IP^{-1} \bullet T_1 \bullet T_2 \bullet \dots \bullet T_{15} \bullet T_{16} \bullet IP(c)$$

- DES的解密过程与加密过程十分相似
- 不同是，将16次迭代的顺序颠倒
- 可以证明

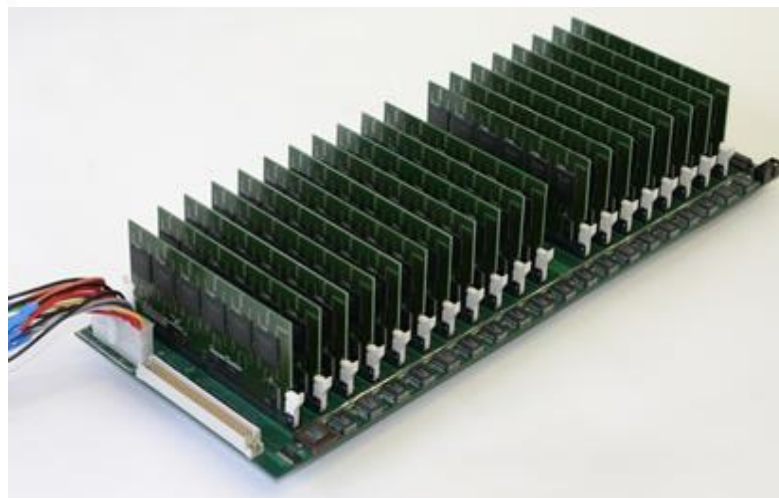
$$DES^{-1}(DES(m)) = m$$



DES算法

- DES的安全性

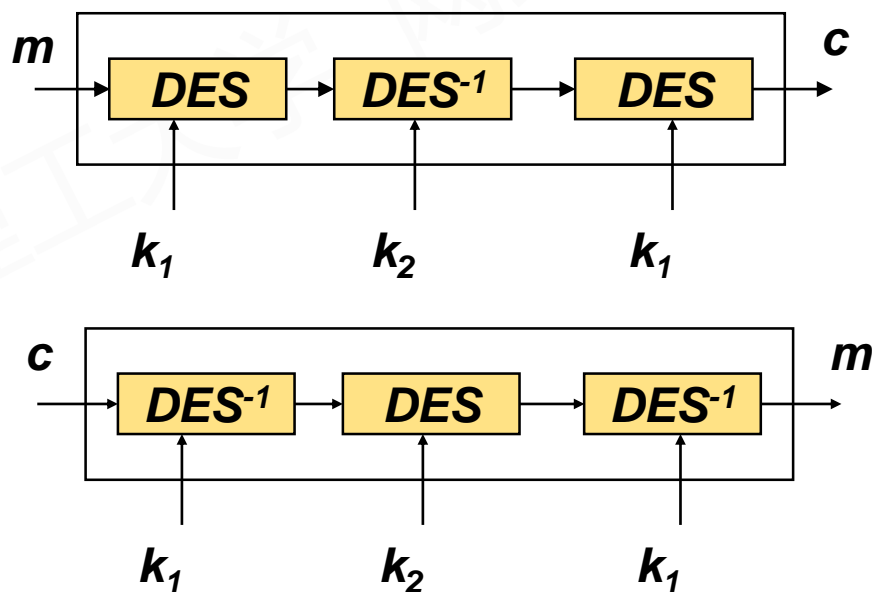
- 1976年，耗资2000万美元的计算机，可以在一天中找到密钥。
- 1993年，设计100万美元的计算机3.5小时用穷举法找到密钥。
- 2006年，1万美元的计算机可以用6.4天可以找到密钥。
- DES安全性不足源自于密钥太短



Triple DES算法

- Triple DES算法

- 3DES算法，密钥长度是112比特， $k=k_1k_2$
- 密钥数量： 2^{112}



DES算法的硬件实现

- 商业DES芯片

- VLSI公司的VM009芯片
- 1993年制造, 200M Bytes/s
- 2009年, 采用FPGA技术, 速度可达到750M Bytes/s

- 软件实现

- 采用双核服务器, 速度约19M Bytes/s

对称密钥密码算法

算法	密钥长度	迭代次数	数学操作	应用
DES	56	16	XOR, S-Box	
3DES	112 or 168	48	XOR, S-Box	PGP, S/MIME
IDEA	128	8	XOR, +, ×	PGP
BlowFish	最大448	16	XOR, S-Box, +	
RC5	最大2048	<255	+, −, XOR	
CAST-128	40—128	16	+, −, S-Box	PGP

RC4算法



- 算法历史

- RC4是RSA算法发明人Ronald Rivest在1987年设计
- 2002年图灵奖获得者
- Ron's Code 4 (Rivest Cipher 4), RC2, RC5, RC6
- 密钥长度可变的流加密算法簇
- 其核心部分的S-Box长度可为任意，但一般为256字节
- 该算法速度是DES算法的10倍

RC4算法

- 算法思想

- RC4产生一个伪随机bit流，与输入流明文逐位异或
- 解密时，用密文与伪随机bit流逐位异或产生明文
- 异或操作具有对称性：

$$0 \times 1 \times 1 = 0 ; 0 \times 0 \times 0 = 0 ; 1 \times 1 \times 1 = 1 ; 1 \times 0 \times 0 = 1$$

- RC4算法关键是根据密钥产生伪随机bit流
 - 使用了一个256位的转换器（S-BOX）
 - 两个指针

RC4算法

- 密钥的使用（密钥调度算法）

- 对于变长密钥，应用它置换S-BOX
- 伪代码，其中S为一个字节数组

```
for i from 0 to 255
    S[i] := i
endfor
j := 0
for i from 0 to 255
    j := (j + S[i] + key[i mod keylength]) mod 256
    swap(&S[i], &S[j])
endfor
```

RC4算法

- 伪随机流产生算法

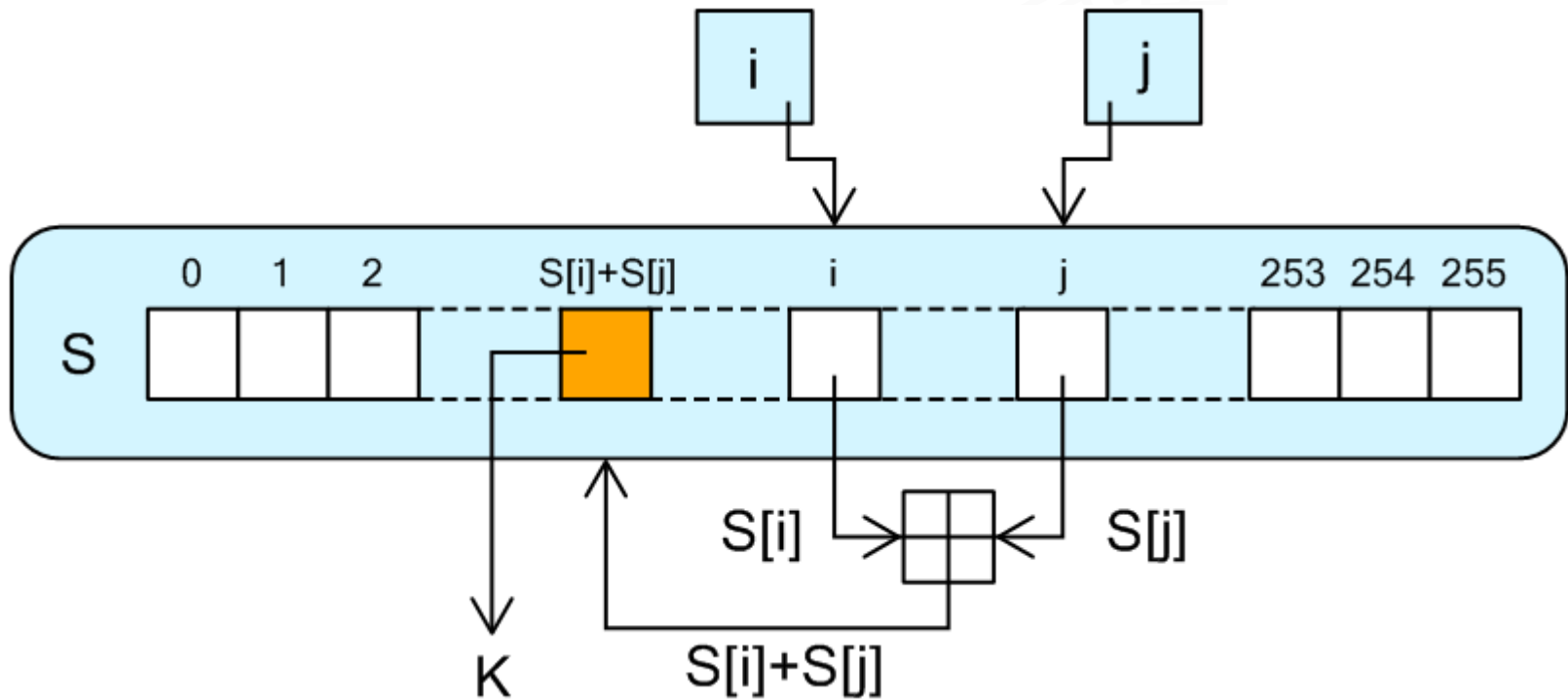
- RPGA: pseudo-random generation algorithm

- 每周期*i*增加1, *j*增加*S[i]*, 每次运算可输出一个Byte

```
i := 0
j := 0
while GeneratingOutput:
    i := (i + 1) mod 256
    j := (j + S[i]) mod 256
    swap(&S[i], &S[j])
    output S[(S[i] + S[j]) mod 256]
endwhile
```

RC4算法

- 伪随机流产生算法



本节小结

- 经过本节的学习，我们知道
 - 密码学悠久的历史
 - 密码学的分类
 - 对称密钥密码算法
 - DES、3DES
 - RC4算法
- 请大家复习课件内容