山东大学
SHANDONG UNIVERSITY

网络与大数据安全

2 - Cryptography

李琨
Email: kunli@sdu.edu.cn

学无止境 气有浩然

山东大学 SHANDONG UNIVERSITY

目录

CONTENTS

1.密码学发展史

2.古典密码学

3.DES与AES

4.密钥交换与公钥密码学

5.RSA与ECC

6.数字签名与MAC

回顾

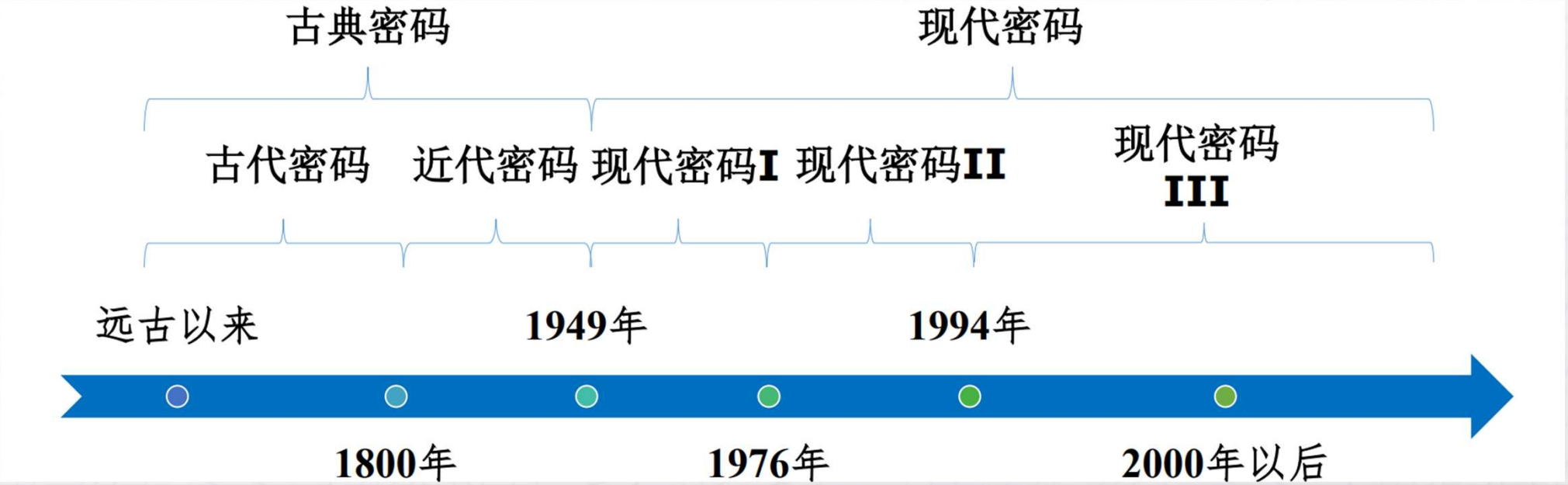
- 安全概念：
  - 机密性；完整性；可用性
  - 真实性；保障性；匿名性
- 加密工具概述
  - 对称/非对称（公钥）加密，哈希，数字签名，数字证书。
- 安全密码
  - 身份认证的常用方法
  - 通常采用哈希存储
  - long psw. + odd char. are better and safer
- 实验
  - 制作数字证书
  - 哈希算法
  - 凯撒密码与维吉尼亚密码

密码学发展史

密码学发展史

密码是一个既古老又现代的名词

- 密码学(Cryptology) 源自希腊文“krypto‘s”及“logos”两字，直译即为“隐藏”及“讯息”之意。
- 人类使用密码的历史非常悠久。自从有了部落，就有了外交，有了战争，也就有了密码
- 人类使用密码的历史几乎与使用文字的时间一样长，戴维·卡恩（David Kahn） 著《破译者》
- 古时候的密码主要是用于军事和外交目的，自古以来，密码都是决定战争胜负的关键
- 密码的起源，密码作为一个学科之前



原始符号——斐斯托斯圆盘

直径约160毫米的粘土圆盘，始于公元前17世纪。表面可有象形符号，有人认为这些符号记载的是某种古代天文历法，但至今还没有任何学者能够解读这些符号的意义

- **1908**年希腊克里特岛的斐斯托皇宫遗址发现，现存于希腊的伊拉克利翁考古博物馆
- 圆盘上的符号是用活字印模在泥盘尚湿时压印上去的，正反两面共可有**241**个象形符号，由外向内螺旋排布，表示了人物、动物、植物、工具等**45**种不同的事物

宗教符号

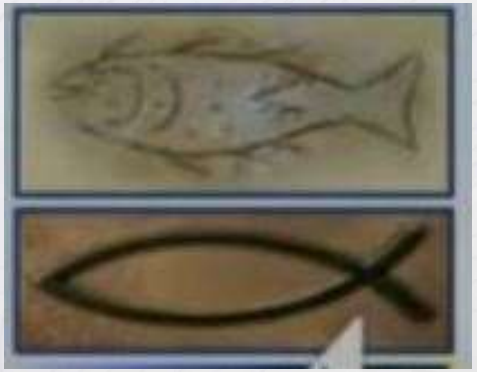
随着人类社会的发展，专制政治统治下，人们发明了各种宗教符号，来通过隐秘的方式传达信仰。其中，遭受迫害的非正统宗教及社团尤为突出，往往试图借鉴典籍来制定密语，有些宗教符号的影响延续至今。

罗马帝国，基督教作为地下教派，常遭到当局迫害。因此基督教徒发明了秘密符号来表达信仰

鸽子—圣灵
孔雀--不朽复活

身背羊羔的牧羊人
守护子民的耶稣

锚—十字架
暴风雨后的安定祥和

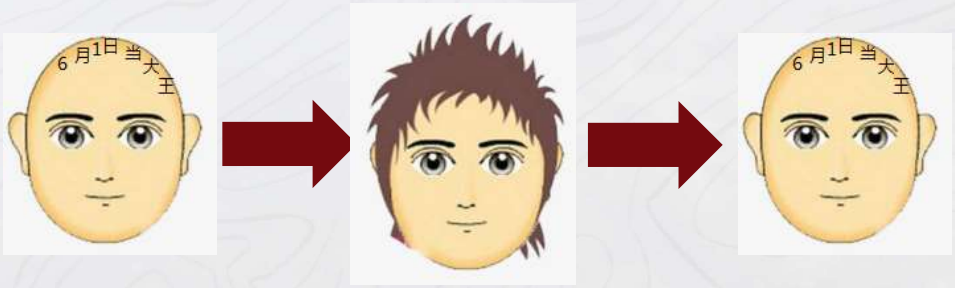


鱼和捕鱼者—永生
鱼--繁殖生命延续

消息隐藏

- 尤其在战争或国家安全中应用，可追溯到几个世纪以前。
- 早在字母表和数字编码发明之前，人们开发出精妙的技巧隐藏消息，一些方法沿用至今
- 这些方法尽管不是非常严格的加密过程，但是其主旨与密码学的目标是相同的，而如今称为“隐写术”

公元前440年，在古希腊战争中，为了安全地传送军事情报，奴隶主剃光奴隶的头发，将情报写在奴隶的光头上，待头发长起后将奴隶送到另一个部落，从而实现了这两个部落之间的秘密通信。

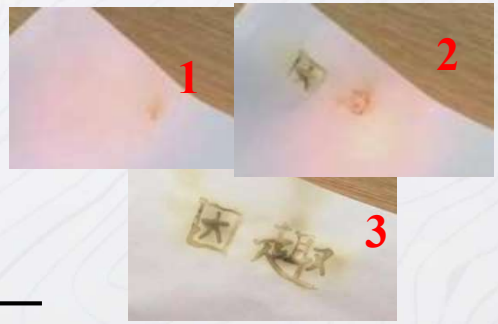


➢ 物理变化（牛奶、蜡笔）

使用步骤：

1. 写信人用牛奶在纸上书写秘密的信息。
2. 接信人在写了秘密信息的纸背面加热，可以看到密文。

牛奶受热的速度比纸张要慢，所以我们可以看到涂有牛奶的地方和周围的白纸有明显的不同



使用步骤：

1.在白纸上用白色的蜡笔写字。

2.用毛笔刷上一层水彩。因为蜡笔不沾水，文字会重新显形。

➢ 化学变化（淀粉、碘酒）

使用步骤：

1.取棉签，蘸上兑米汤，将要写的内容写在一张白纸上。晾干。

3.用毛笔蘸上碘酒刷在之前写字的地方，等待字体显现。

➢ 不用墨水

这一方法主要利用书写时用力所产生的刻痕。

1.拿出两张纸。将第一张覆盖在第二张纸上方。

2.将文字用力写在第一张纸上。

3.拿掉第一张纸。

4.在第二张纸的刻痕处用铅笔轻轻地打上阴影。铅笔会掠过刻痕，文字就如同在黑板上的粉笔字一样明显。

密码学发展史

- 古典密码：1949年前
  - 手工密码：以手工完成加密作业，或者以简单器具辅助操作的密码，叫作手工密码。第一次世界大战前主要是这种作业形式。
    - 埃及人、希伯来人、亚述人、希腊人
    - Scytale
    - 凯撒密码
    - 维吉尼亚密码
  - 机械密码：以机械密码机或电动密码机来完成加解密作业的密码，叫作机械密码。这种密码从第一次世界大战出现到第二次世界大战中得到普遍应用。
    - 单表代替、多表代替、转轮密码

斯巴达密码棒

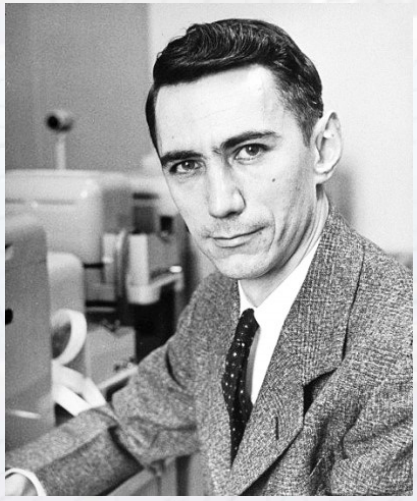
明文	A	B	C	D	E	F	G	H	I	J	K	L	M
密文	F	G	H	I	J	K	L	M	N	O	P	Q	R
明文	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
密文	ST	U	V	W	X	Y	Z	A	B	C	D	E	F
	S	T	U	V	W	X	Y	Z	A	B	C	D	E

凯撒密码明密文对应表

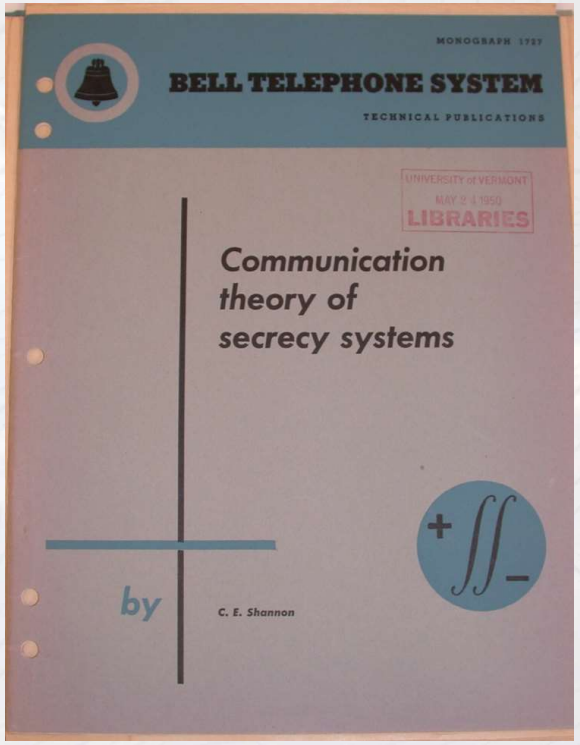
Enigma密码机（1918）

密码学发展史

- 随着信息社会的到来，大量信息需要加密，于是人们便开始密码技术的研究。
- 20年代就出现了加密机和密码研究的文献。
- 1949年Shannon发表了“保密系统的通信理论”的论文，奠定了私钥密码学的理论基础，使密码学成为一门科学。
- 但由于当时密码研究主要还是在军事领域进行，密码人员都在黑屋子里工作，处于保密状态。所以1949——1976年近20年里密码的理论和技术没有大的进展。



香农



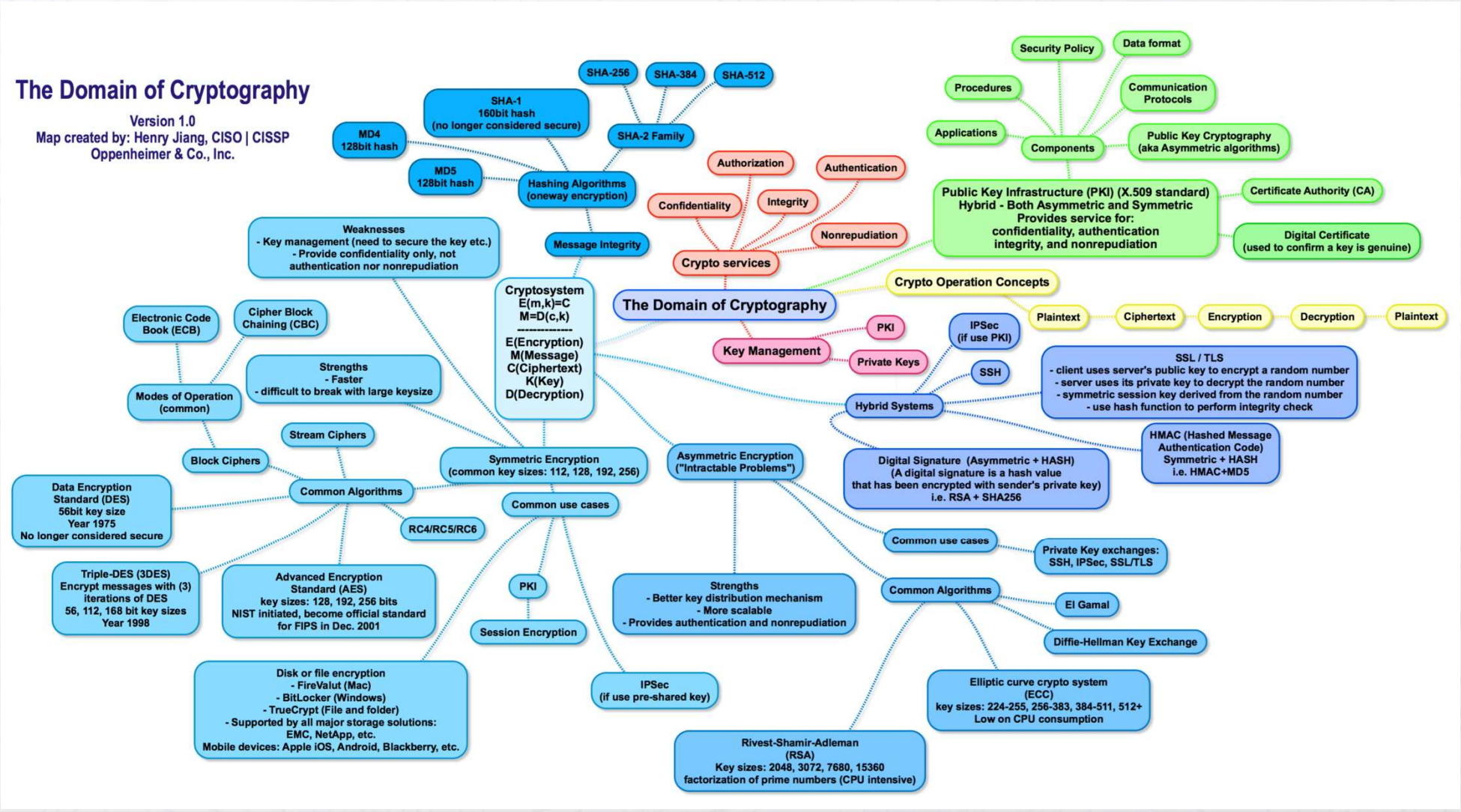
保密系统的通信理论

Shannon的信息论：只有密钥长度与明文长度一样的加密才是绝对安全的——即一次一密

密码学发展史

- 直到70年代，密码学界出现了两个重大事件：
  - 1976年，Diffie and Hellman 发表了题为“密码学的新方向”一文，提出了一种新的密码体制-----公钥密码体制
  - 1977年，美国正式公布实施数据加密标准DES，算法完全公开并广泛用于商业数据加密。
- 从此，密码学的研究高潮迭起，密码技术迅速发展。
- 1997年4月15日美国国家标准技术研究所（NIST）发起了征集高级加密标准算法A（advanced）ES的活动
- 现在美国新的高级加密标准已经出台，欧洲也耗巨资完成了“欧洲密码大计划”，制定了自己的标准。
- 现代密码（电子芯片、计算机）
- 未来的密码（量子密码、混沌密码、DNA密码……）

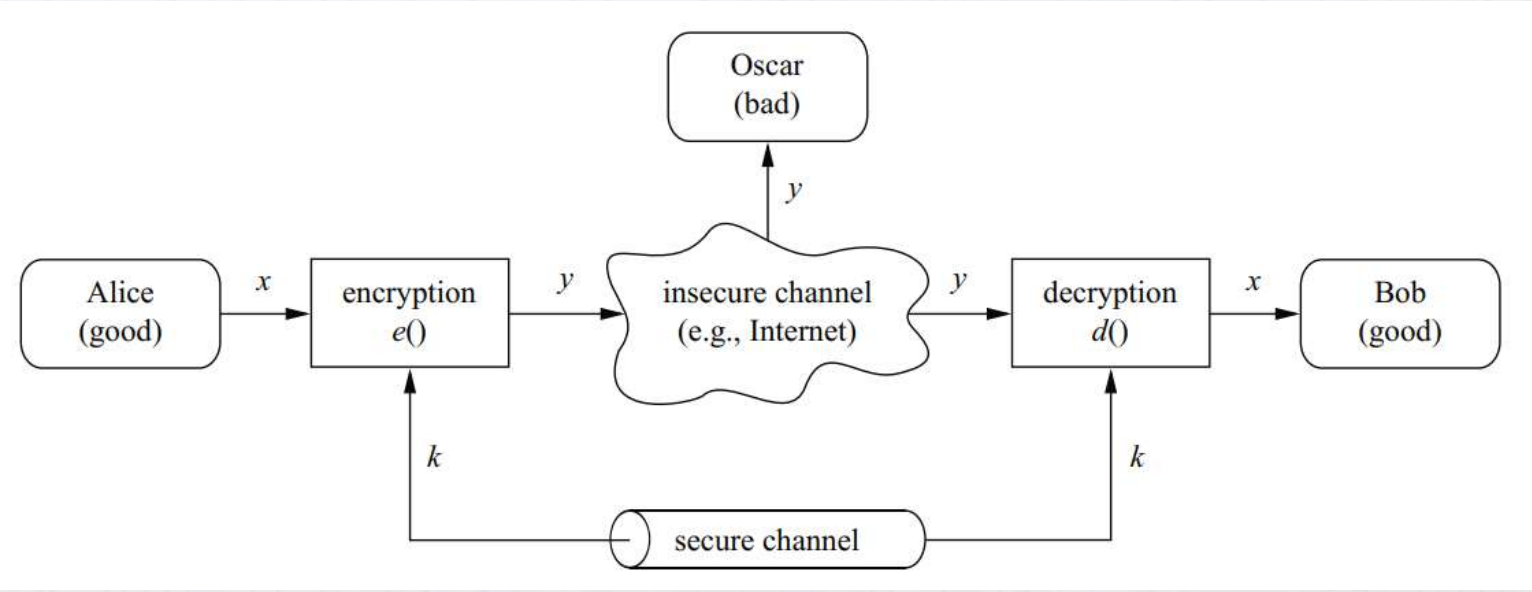
密码学发展史


The Domain of Cryptography
Version 1.0
Map created by: Henry Jiang, CISO | CISSP
Oppenheimer & Co., Inc.
The Domain of Cryptography
Cryptosystem
E(m,k)=C
M=D(c,k)
E(Encryption)
M(Message)
C(Ciphertext)
K(Key)
D(Decryption)
Crypto services
Confidentiality
Integrity
Authorization
Authentication
Nonrepudiation
Message Integrity
Hashing Algorithms
(oneway encryption)
SHA-1
160bit hash
(no longer considered secure)
SHA-2 Family
SHA-256
SHA-384
SHA-512
MD4
128bit hash
MD5
128bit hash
Weaknesses
- Key management (need to secure the key etc.)
- Provide confidentiality only, not
authentication nor nonrepudiation
Public Key Infrastructure (PKI) (X.509 standard)
Hybrid - Both Asymmetric and Symmetric
Provides service for:
confidentiality, authentication
integrity, and nonrepudiation
Components
Security Policy
Data format
Procedures
Communication
Protocols
Applications
Public Key Cryptography
(aka Asymmetric algorithms)
Certificate Authority (CA)
Digital Certificate
(used to confirm a key is genuine)
Crypto Operation Concepts
Plaintext
Ciphertext
Encryption
Decryption
Plaintext
Key Management
PKI
Private Keys
Hybrid Systems
IPSec
(if use PKI)
SSH
SSL / TLS
- client uses server's public key to encrypt a random number
- server uses its private key to decrypt the random number
- symmetric session key derived from the random number
- use hash function to perform integrity check
HMAC (Hashed Message
Authentication Code)
Symmetric + HASH
i.e. HMAC+MD5
Digital Signature (Asymmetric + HASH)
(A digital signature is a hash value
that has been encrypted with sender's private key)
i.e. RSA + SHA256
Symmetric Encryption
(common key sizes: 112, 128, 192, 256)
Modes of Operation
(common)
Electronic Code
Book (ECB)
Cipher Block
Chaining (CBC)
Strengths
- Faster
- difficult to break with large keysize
Stream Ciphers
Block Ciphers
Common Algorithms
Data Encryption
Standard (DES)
56bit key size
Year 1975
No longer considered secure
Triple-DES (3DES)
Encrypt messages with (3)
iterations of DES
56, 112, 168 bit key sizes
Year 1998
Advanced Encryption
Standard (AES)
key sizes: 128, 192, 256 bits
NIST initiated, become official standard
for FIPS in Dec. 2001
RC4/RC5/RC6
Common use cases
PKI
Session Encryption
IPSec
(if use pre-shared key)
Disk or file encryption
- FireValut (Mac)
- BitLocker (Windows)
- TrueCrypt (File and folder)
- Supported by all major storage solutions:
EMC, NetApp, etc.
Mobile devices: Apple iOS, Android, Blackberry, etc.
Asymmetric Encryption
("Intractable Problems")
Strengths
- Better key distribution mechanism
- More scalable
- Provides authentication and nonrepudiation
Common use cases
Private Key exchanges:
SSH, IPSec, SSL/TLS
Common Algorithms
El Gamal
Diffie-Hellman Key Exchange
Elliptic curve crypto system
(ECC)
key sizes: 224-255, 256-383, 384-511, 512+
Low on CPU consumption
Rivest-Shamir-Adleman
(RSA)
Key sizes: 2048, 3072, 7680, 15360
factorization of prime numbers (CPU intensive)

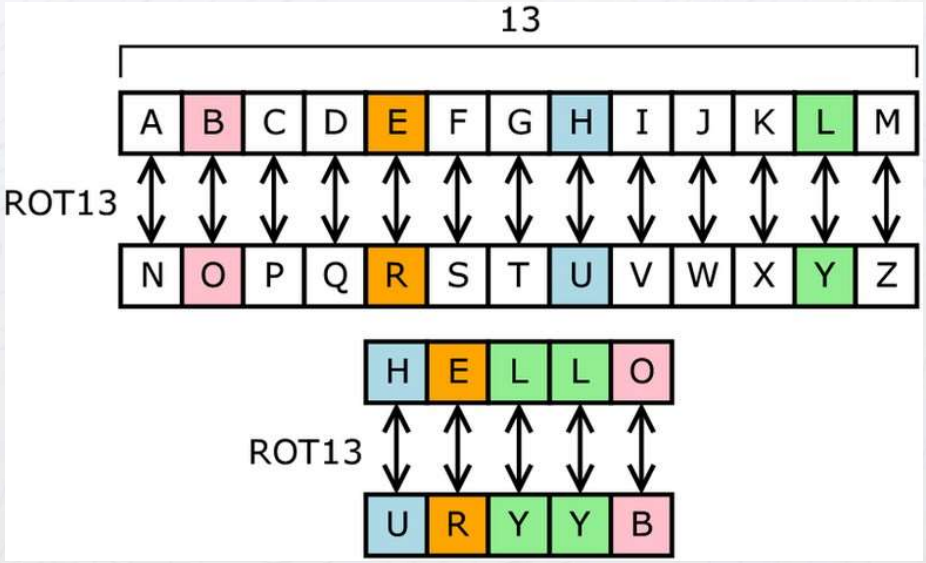

Any Questions?

古典密码学

- 场景
  - Alice 想要向 Bob 发送一条信息（明文 P）.
  - 信道不安全，可能被窃听
  - 爱丽丝和鲍勃事先商定了加密方案和秘钥 K，信息可以加密（密文 C）发送
- 问题
  - 什么是好的对称加密方案?
  - 加密/解密的复杂度是多少?
  - 相对于明文，密文的大小是多少?



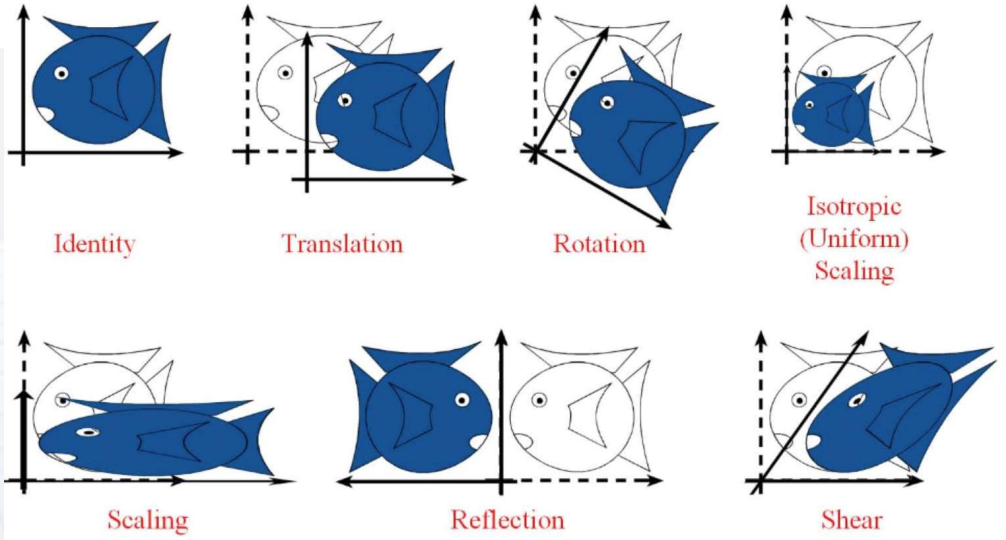
- 代换加密
  - 将每个字母替换成另一个字母。
  - 单表代换：对明文消息中出现的同一个字母，在加密时都使用同一固定的字母来代换，不管它出现在什么地方。
  - 多表代换：明文消息中出现的同一个字母，在加密时不是完全被同一固定的字母代换，而是根据其出现的位置次序，用不同的字母代换。



仿射密码

- 仿射变换：线性变换+平移
  - 变换前是直线的，变换后依然是直线。
  - 直线比例保持不变。
  - 包括平移、旋转、缩放、反射、错切。

$$f(x) = Ax + b, \quad x \in X$$



- 模运算：取余

  整数 a 除以整数 b，若得到的余数是 r，则记作$a\ mod\ b = r$，例如$5\ mod\ 3 = 2$, $-5\ mod\ 3 = 1$

- 同余

  两个整数 a, b，若它们除以正整数 n 所得的余数相等，即 $a\ mod\ n = b\ mod\ n$，则称 a 和 b 对于模 n 同余，记作$a \equiv b(mod\ n)$，例如$2 \equiv 8(mod\ 6)$

- 模逆元

  对整数 a 和 b，若$ab \equiv 1(mod\ n)$，则称 a 和 b 关于模 n 互为模倒数，也称模逆元或模反元素，还可以记作$b \equiv \frac{1}{a}(mod\ n)$或$b \equiv a^{-1}\ (mod\ n)$，例如$5 \equiv \frac{1}{3}(mod\ 7)$

仿射密码

- 仿射密码：
  - $x, y, a, b \in Z_{26}$
  - 加密：$e_k(x) = y \equiv a \cdot x + b \ \ mod\ 26.$
  - 解密：$d_k(y) = x \equiv a^{-1} \cdot (y - b) \ \ mod\ 26.$
  - 密钥：$k = (a, b), \gcd(a, 26) = 1.$
- 示例：
  - 密钥$k = (7, 3)$
  - 加密函数是$e_k = 7x + 3\ mod\ 26$
  - 解密函数是$d_k = 15(y - 3) mod\ 26 = 15y\ - 19\ mod\ 26$

计算7 mod 26 的逆元：$7x = 1\ mod\ 26$
转化为$7x - 26y = 1$

$$26 = 3 \times 7 + 5$$
$$7 = 1 \times 5 + 2$$
$$5 = 2 \times 2 + 1$$
$$2 = 1 \times 2$$

由此可得：$1 = 5 - 2 \times 2$
$= 5 - 2 \times (7 - 5) = 3 \times 5 - 2 \times 7$
$= 3 \times (26 - 3 \times 7) - 2 \times 7$
$= 3 \times 26 - 11 \times 7 = 7 \times (-11) - 26 \times (-3)$
由于我们希望找到正的逆元，可以将-11转换为模26下的等价正数：15

明文	h	o	t
x	7	14	19
$7x + 3\ mod\ 26$	0	23	6
密文	a	x	g

密文	a	x	g
y	0	23	6
$15y - 19\ mod\ 26$	7	14	19
明文	h	o	t

仿射密码

- 仿射密码：
  - $x, y, a, b \in Z_{26}$
  - 加密: $e_k(x) = y \equiv a \cdot x + b \ \ mod\ 26.$
  - 解密: $d_k(y) = x \equiv a^{-1} \cdot (y - b) \ \ mod\ 26.$
  - 密钥: $k = (a, b), \gcd(a, 26) = 1.$
- 示例：
  - 密钥$k = (7, 3)$
  - 加密函数是$e_k = 7x + 3\ mod\ 26$
  - 解密函数是$d_k = 15(y - 3) mod\ 26 = 15y\ - 19\ mod\ 26$

模26的所有逆元组合

1： 1
3： 9
5： 21
7： 15
11： 19
17： 23
25： 25

练习：给定密钥(5,8)，请写出其仿射密码对应的加密函数与解密函数，并对AFFINECIPHER进行加密。

仿射密码

- 仿射密码：
  - $x, y, a, b \in Z_{26}$
  - 加密: $e_k(x) = y \equiv a \cdot x + b \ mod\ 26.$
  - 解密: $d_k(y) = x \equiv a^{-1} \cdot (y - b) \ mod\ 26.$
  - 密钥: $k = (a, b), \gcd(a, 26) = 1.$
- 示例：
  - 密钥$k = (7, 3)$
  - 加密函数是$e_k = 7x + 3 \ mod\ 26$
  - 解密函数是$d_k = 15(y - 3) mod\ 26 = 15y\ - 19 \ mod\ 26$

模26的所有逆元组合

1： 1
3： 9
5： 21
7： 15
11： 19
17： 23
25： 25

练习：给定密钥(5,8)，请写出其仿射密码对应的加密函数与解密函数，并对AFFINECIPHER进行加密。

加密： $5x + 8 \ mod\ 26$ 解密： $21(y - 8) mod\ 26 = 21y - 12 \ mod\ 26$

密文：IHHWVCSWFRCP

- 置换加密

是指明文中各字符的位置次序重新排列得到密文的一种密码体制。
特点：保持明文中所有的字符不变，只是利用置换打乱明文字符的位置和次序。
例：栅栏密码

➢ 置换

一个有限集合 S 到自身的双射（即一一对应）称为 S 的一个置换。集合 $S=\{a_1,a_2,\ldots,a_n\}$ 上的置换可以表示为

$$f=\begin{pmatrix} a_1,a_2,\ldots,a_n \\ a_{p_1},a_{p_2},\ldots,a_{p_n} \end{pmatrix}$$

意为将 a_i 映射为 a_{p_i}，其中 $p_1,p_2,\ldots,p_n$ 是 $1,2,\ldots,n$ 的一个排列。显然 S 上所有置换的数量为 $n!$。

$$\alpha=\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 6 & 4 & 7 & 2 & 5 & 1 & 8 & 9 & 3 \end{pmatrix} \Longrightarrow \alpha=(1\ 6)(2\ 4)(3\ 7\ 8\ 9)(5)$$

- 置换加密

是指明文中各字符的位置次序重新排列得到密文的一种密码体制。
特点：保持明文中所有的字符不变，只是利用置换打乱明文字符的位置和次序。
例：栅栏密码

➢ 周期置换

1. 将明文串P按固定长度m进行分组
2. 对每组中的子串按1，2…，m的某个置换重排位置从而得到密文C

例：明文 State Key Laboratory of Networking and Switching，假定m为6，密钥$\sigma = (15623)$，则分组后的明文是
(StateK)(eyLabo)(ratory)(ofNetw)(orking)(andSwi)(tching)
加密变换：(akttSe)(Loyaeb)(tyaorr)(Nwfeot)(kgrion)(dinSaw)(hgcitn)
最终密文：akttSeLoyaebtyaorrNwfeotkgriondinSawhgcitn
可推断出解密密钥是(13265)

- 置换加密

是指明文中各字符的位置次序重新排列得到密文的一种密码体制。
特点：保持明文中所有的字符不变，只是利用置换打乱明文字符的位置和次序。
例：栅栏密码

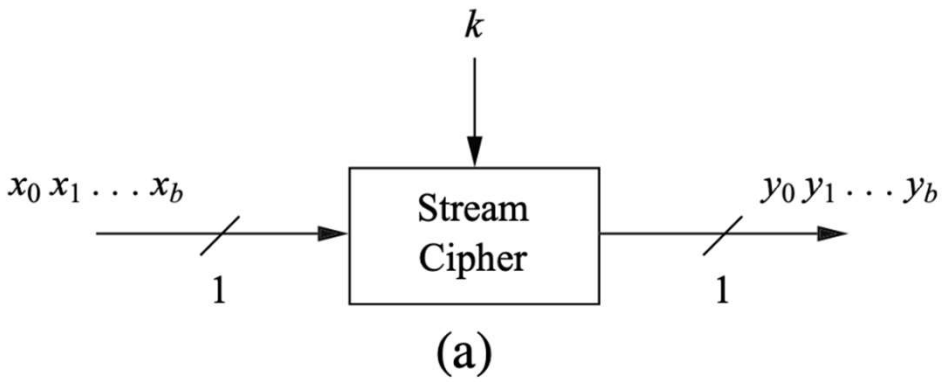
➢ 列置换

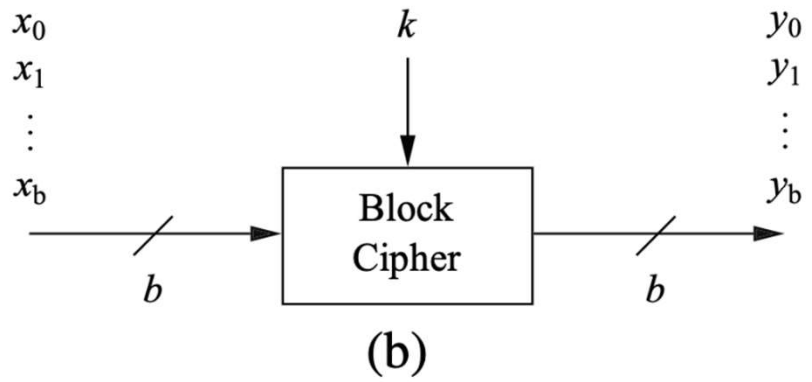
1. 将明文P以设定的固定分组宽度m按行写出，即每行有m个字符。若明文长度不是m的整数倍，则不足部分用双方约定的方式填充，将排列好的字符矩阵记作$[M]_{m*n}$。
2. 按$1,2\ldots,m$的某一置换σ交换列的位置次序得字符矩阵 $[M_P]_{m*n}$
3. 把矩阵$[M_P]_{m*n}$按列$1,2\ldots,n$的顺序依次读出得密文序列C

例：明文TheXXIVOlympicWinterGames，假定m为5，密钥$\sigma=(143)(25)$，密文eOitmXyWrsXlceeTImiGhVpna
可推断出解密密钥是$(134)(25)$

$$[M]_{5*5}=\begin{bmatrix} T & h & e & X & X \\ I & V & O & l & y \\ m & p & i & c & W \\ i & n & t & e & r \\ G & a & m & e & s \end{bmatrix} \overset{\sigma}{\Rightarrow} [M_P]_{5*5}=\begin{bmatrix} e & X & X & T & h \\ O & y & l & I & V \\ i & W & c & m & p \\ t & r & e & i & n \\ m & s & e & G & a \end{bmatrix}$$

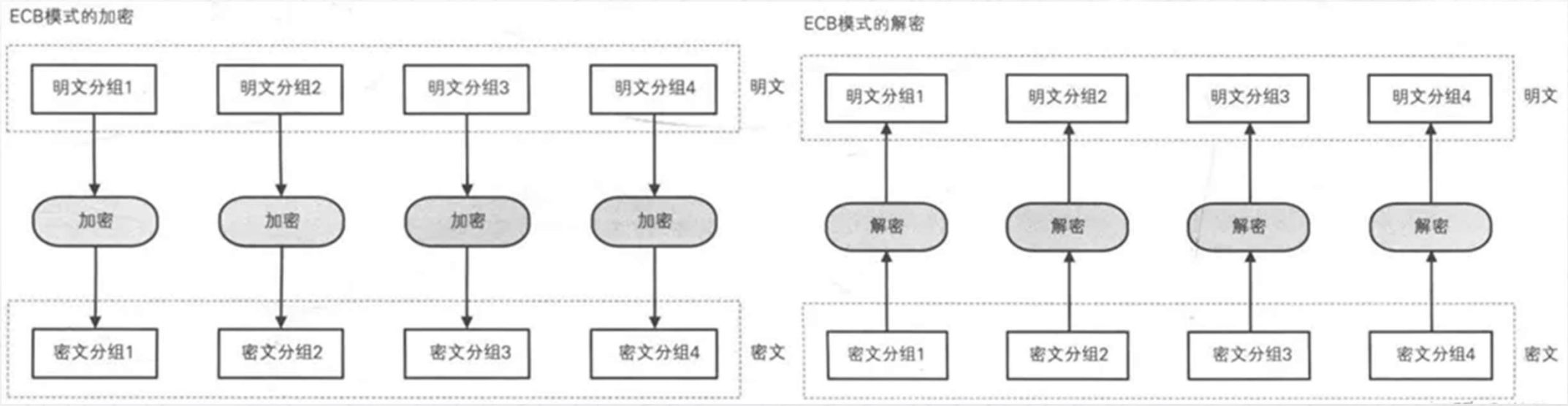
- 流密码与分组密码



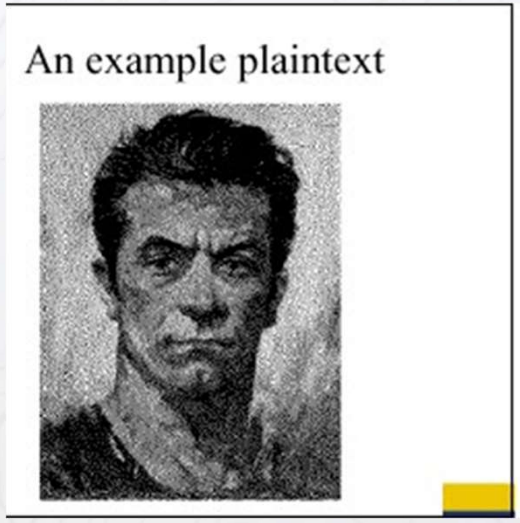


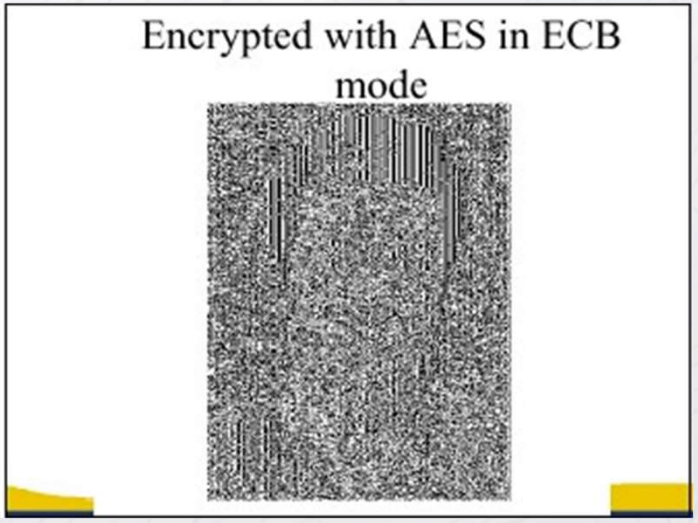
- 流密码也称为序列密码，是一种单钥体制的加密方式。在流密码中，明文消息被逐位地加密，加密和解密双方使用相同的密钥流。
- 分组密码是一种将明文消息分组（含有多个字符）后逐组进行加密的密码体制。在分组密码中，将大小为m的一组明文符号作为整体进行加密，创建出相同大小的一组密文。

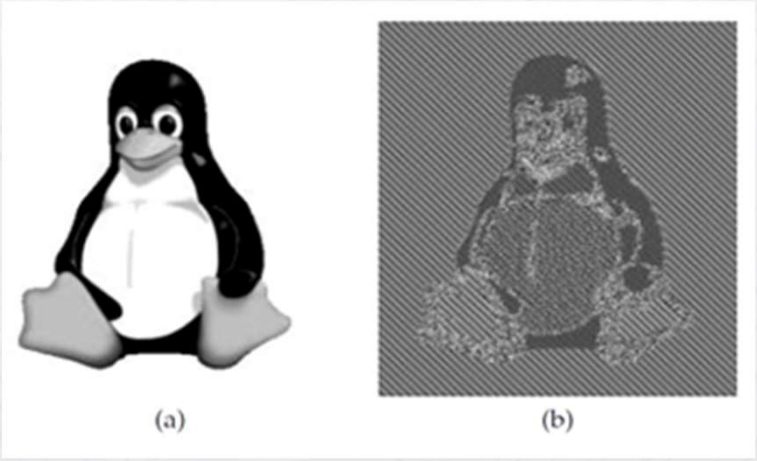
- 分组密码的主要模式
  - ECB模式(Electronic CodeBook)：将明文分组加密之后的结果将直接成为密文分组。
    - 特点：相同的明文分组会被转换为相同的密文分组（一一对应），因此ECB模式也称为**电子密码本模式**。
    - 攻击：攻击者只要对任意密文分组进行替换、删除或复制，相应的明文分组也会被替换、删除或复制，即**攻击者无需破译密码就能够操纵明文**。



- 分组密码的主要模式
  - ECB模式(Electronic CodeBook)：将明文分组加密之后的结果将直接成为密文分组。
    - 特点：相同的明文分组会被转换为相同的密文分组（一一对应），因此ECB模式也称为**电子密码本模式**。
    - 攻击：攻击者只要对任意密文分组进行替换、删除或复制，相应的明文分组也会被替换、删除或复制，即**攻击者无需破译密码就能够操纵明文**。
    - 不适用于图像加密





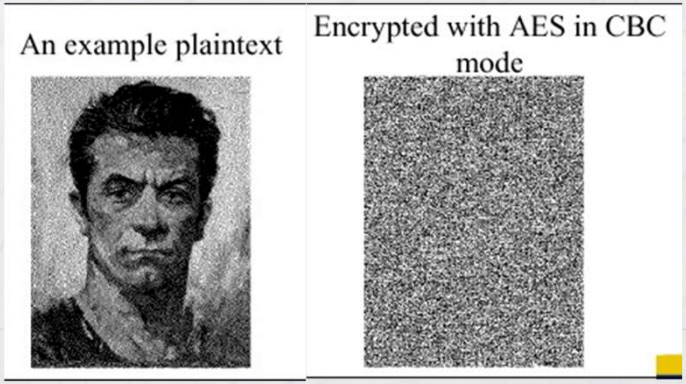


- 分组密码的主要模式
  - CBC模式(Cipher Block Chaining)：首先将明文分组与前一个密文分组进行XOR运算，然后再进行加密。
  - 特点：
    - a. 明文分组在加密之前一定会与"前一个密文分组"进行XOR运算，因此即使明文分组1和2的值是相等的，密文分组的值也不一定是相等的。
    - b. 在加密过程中，我们无法单独对一个中间的明文分组进行加密。
    - c. 在解密过程中，假设有一个密文分组损坏了，只要密文分组的长度没有变化，则解密时最多只会有2个分组受到数据损坏的影响。
    - d. 假设密文分组中有一些比特缺失了，那么会导致密文分组的长度发生变化，此后的分组发生错位，则缺失比特的位置之后的密文分组就全部无法解密了。

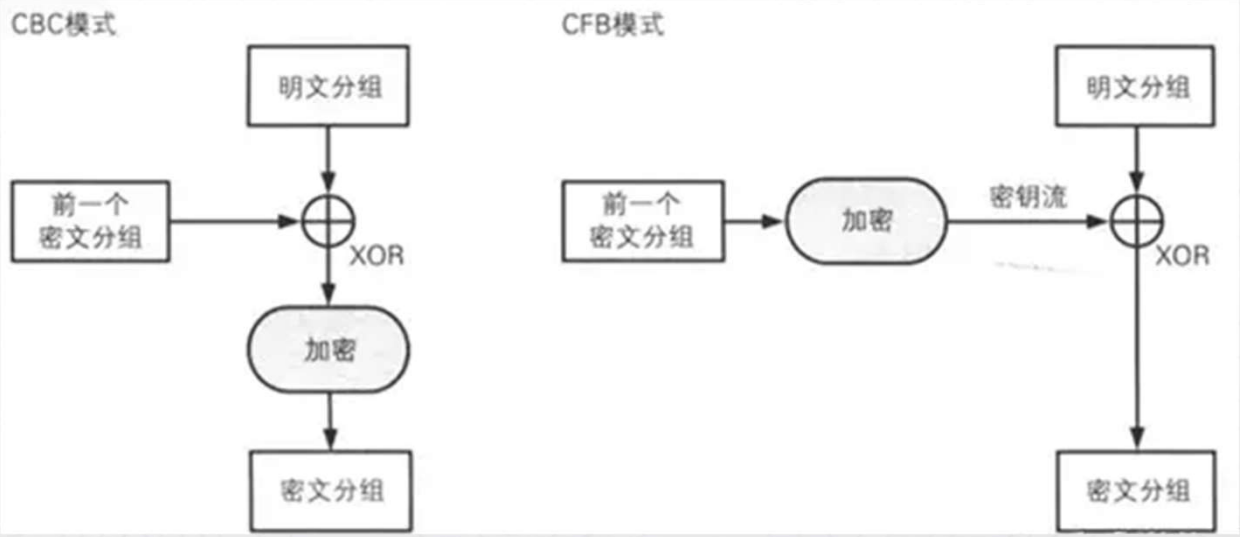
填充提示攻击是一种利用分组密码中的填充部分来进行攻击的方法，**适用于所有需要进行分组填充的模式。**（在分组密码中，当明文长度不为分组长度的整数倍时，需要在最后一个分组中填充一些数据使其凑满一个分组长度。）攻击者会反复发送一段密文，**每次发送时都对填充的数据进行少许改变。**由于接收者在无法正确解密时会返回一个错误消息，攻击者通过这一错误消息就可以获得一部分与明文相关的信息。**防御这种攻击的方法是对密文进行认证，以确保密文是由合法的发送者在知道明文内容的前提下生成的。**

异或XOR：如果两个值不相同，则异或结果为1。如果两个值相同，异或结果为0。

- 分组密码的主要模式
  - CBC模式(Cipher Block Chaining)：首先将明文分组与前一个密文分组进行XOR运算，然后再进行加密。
  - 特点：
    - a. 明文分组在加密之前一定会与"前一个密文分组"进行XOR运算，因此即使明文分组1和2的值是相等的，密文分组的值也不一定是相等的。
    - b. 在加密过程中，我们无法单独对一个中间的明文分组进行加密。
    - c. 在解密过程中，假设有一个密文分组损坏了，只要密文分组的长度没有变化，则解密时最多只会有2个分组受到数据损坏的影响。
    - d. 假设密文分组中有一些比特缺失了，那么会导致密文分组的长度发生变化，此后的分组发生错位，则缺失比特的位置之后的密文分组就全部无法解密了。



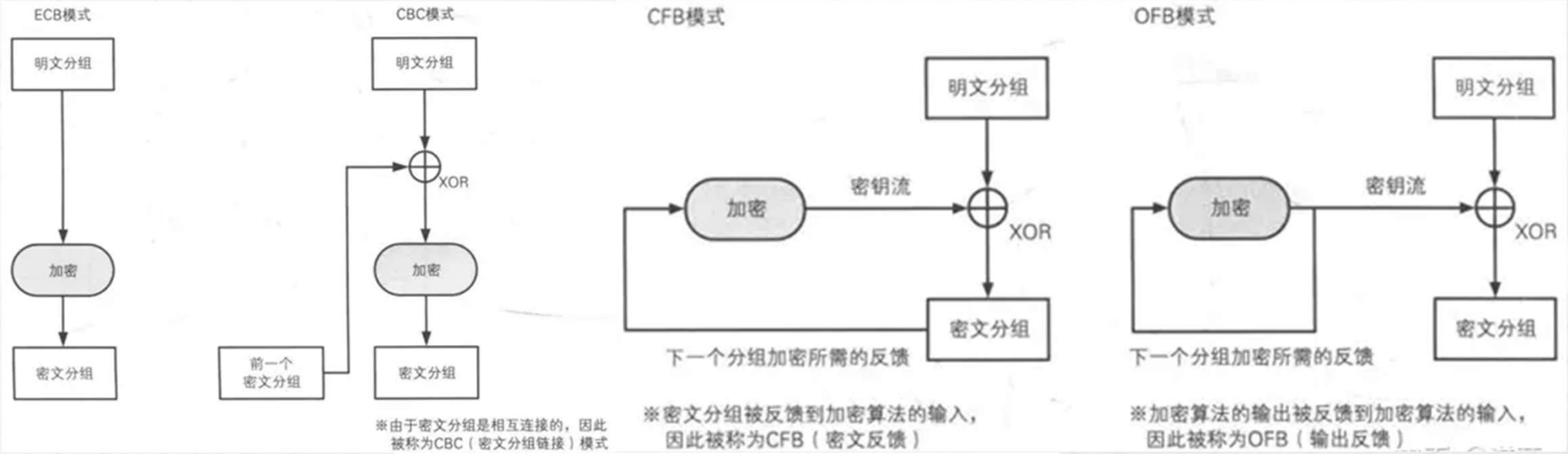
- 分组密码的主要模式
  - CFB模式(Cipher FeedBack)：前一个密文分组会被送回到密码算法的输入端（即所谓的“反馈”）。
  - 特点：
    - 在CBC模式中，明文分组和密文分组之间有XOR和密码算法两个步骤；而在CFB模式中，明文分组和密文分组之间只有XOR。
    - 在CFB模式中，明文数据可以被逐比特加密，因此我们可以将CFB模式看作是一种使用分组密码来实现流密码的方式。

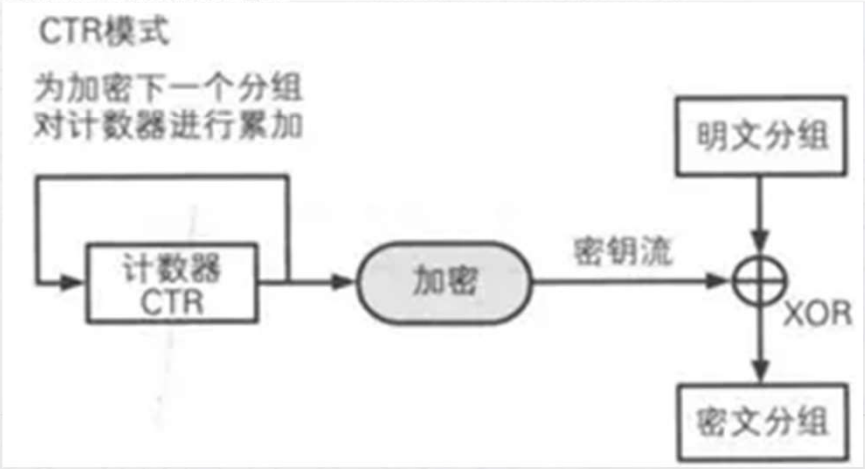


- 分组密码的主要模式
  - OFB模式(Output-FeedBack)：密码算法的输出会反馈到密码算法的输入中。
  - 特点：
    - OFB模式并不是通过密码算法对明文直接进行加密的，而是通过将“明文分组”和“密码算法的输出”进行XOR来产生“密文分组”的。在这一点上OFB模式和CFB模式非常相似。
    - OFB模式和CFB模式的区别仅仅在于密码算法的输入。CFB模式中，密码算法的输入是前一个密文分组（密文反馈模式）；OFB模式中，密码算法的输入则是密码算法的前一个输出（输出反馈模式）。
    - CFB模式是对密文分组进行反馈的，因此无法跳过明文分组1而先对明文分组2进行加密；OFB模式中，XOR所需要的比特序列（密钥流）可以事先通过密码算法生成，和明文分组无关。只要提前准备好所需的密钥流，则在实际加密的过程中，就完全不需要动用密码算法了，只要将明文与密钥流进行XOR即可。因此，生成密钥流的操作和进行XOR运算的操作是可以并行的。

- 分组密码的主要模式
  - CTR模式(CounTeR)：通过将逐次累加的计数器进行加密来生成密钥流的流密码。
  - **计数器**：每次加密时都会生成一个不同的值来作为计数器的初始值。其中前8个字节为nonce（只用一次的随机数），这个值在每次加密时必须都是不同的。后8个字节为分组序号，这个部分是会逐次累加的。
  - **特点**：
    - 加密和解密使用了完全相同的结构，因此在程序实现上比较容易。
    - 可以以任意顺序对分组进行加密和解密，因为计数器的值可以由nonce和分组序号直接计算出来。（**OFB模式不具备**）
    - 能够以任意顺序处理分组，就意味着**能够实现并行计算**。在支持并行计算的系统中，CTR模式的速度是非常快的。

- 分组密码的主要模式


ECB模式
明文分组
加密
密文分组
CBC模式
明文分组
XOR
加密
前一个
密文分组
密文分组
※由于密文分组是相互连接的，因此
被称为CBC（密文分组链接）模式
CFB模式
明文分组
加密
密钥流
XOR
密文分组
下一个分组加密所需的反馈
※密文分组被反馈到加密算法的输入，
因此被称为CFB（密文反馈）
OFB模式
明文分组
加密
密钥流
XOR
密文分组
下一个分组加密所需的反馈
※加密算法的输出被反馈到加密算法的输入，
因此被称为OFB（输出反馈）


CTR模式
为加密下一个分组
对计数器进行累加
明文分组
计数器
CTR
加密
密钥流
XOR
密文分组

模式	名称	优点	缺点
ECB模式	Electronic Code Book电子密码本模式	• 简单快速 • 支持并行计算（加密、解密）	• 明文中的重复排列会反映在密文中 • 通过删除、替换密文分组可以对明文进行操作 • 对包含某些比特错误的密文进行解密时，对应的分组会出错，不能抵御重放攻击
CBC模式	Cipher Block Chaining密文分组链接模式	• 明文的重复排列不会反映在密文中 • 支持并行计算(仅解密） • 能够解密任意密文分组	• 对包含某些错误比特的密文进行解密时，第一个分组的全部比特以及后一个分组的相应比特会出错 • 加密不支持并行计算
CFB模式	Cipher-FeedBack密文反馈模式	• 不需要填充(padding) • 支持并行计算(仅解密） • 能够解密任意密文分组	• 加密不支持并行计算。 • 对包含某些错误比特的密文进行解密时，第一个分组的全部比特以及后一个分组的相应比特会出错 • 不能抵御重放攻击
OFB模式	Output-FeedBack输出反馈模式	• 不需要填充(padding) • 可事先进行加密、解密的准备 • 加密、解密使用相同结构 • 对包含某些错误比特的密文进行解密时，只有明文中相对应的比特会出错	• 不支持并行计算 • 主动攻击者反转密文分组中的某些比特时，明文分组中相对应的比特也会被反转
CTR模式	CounTeR计数器模式	• 不需要填充(padding) • 可事先进行加密、解密的准备 • 加密、解密使用相同结构 • 对包含某些错误比特的密文进行解密时，只有明文中相对应的比特会出错 • 支持并行计算(加密、解密)	• 主动攻击者反转密文分组中的某些比特时，明文分组中相对应的比特也会被反转

- 总结：流密码 V.S. 分组密码
  - 流密码：
    - 连续处理加密/解密元素，每次处理一个比特或字节
    - 优点：序列密码实现简单，便于硬件实施，加解密处理速度快，没有或只有有限的错误传播等特点
    - 缺点：低扩展性，插入及修改的不敏感性，不够安全
  - 分组密码：
    - 一次处理一个元素块的输入并为每个输入块生成一个输出块
    - 优点：明文信息良好的扩展性，对插入的敏感性，不需要密钥同步，较强的适用性
    - 缺点：加密速度慢，错误扩散和传播，穷举法较容易破译

流密码与分组密码可以结合起来同时使用


Any Questions?

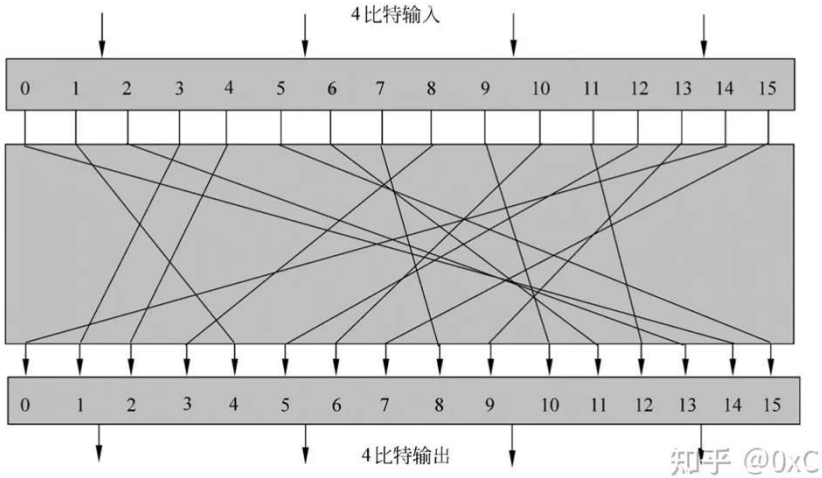
DES与AES

- **SPN结构(代换置换网络 Substitution-permutation networks)**

1949年香农提出

  - 核心思想：复杂的、带有随机性的、重复多轮的分组密码
  - S盒Substitution box（代换）：

    明文的每一个分组都应产生惟一的一个密文分组，我们把这样的变换称为可逆的，称明文分组到密文分组的可逆变换为**代换**。



明文	密文	明文	密文
0000	1101	1000	1010
0001	0101	1001	1000
0010	0001	1010	1011
0011	0110	1011	1110
0100	1001	1100	0111
0101	1111	1101	0100
0110	0010	1110	1100
0111	0011	1111	0000

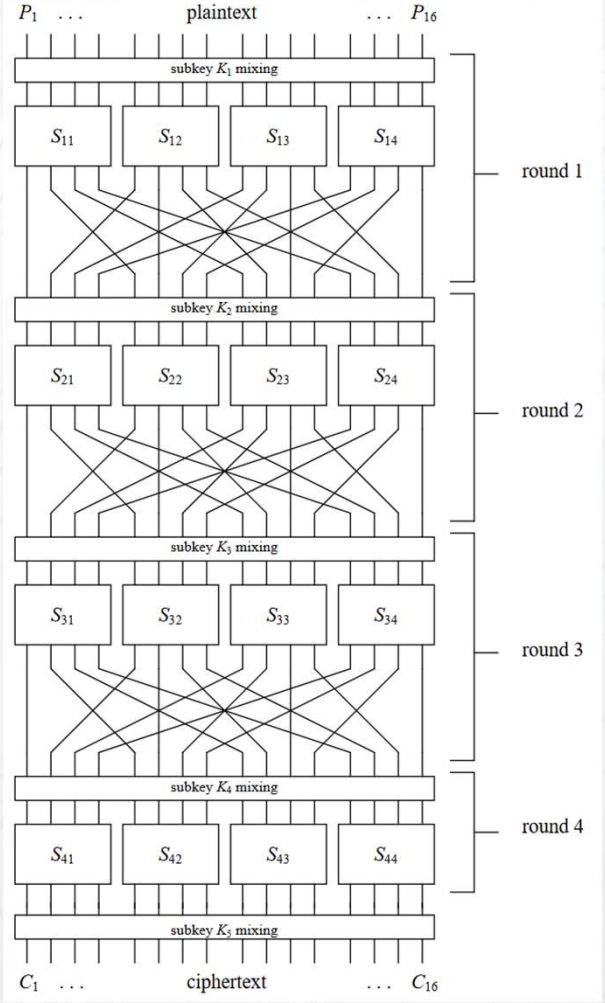
正向代换

密文	明文	密文	明文
0000	1111	1000	1001
0001	0010	1001	0100
0010	0110	1010	1000
0011	0111	1011	1010
0100	1101	1100	1110
0101	0001	1101	0000
0110	0011	1110	1011
0111	1100	1111	0101

反向代换

- **SPN结构(代换置换网络 Substitution-permutation networks)**
  - 核心思想：复杂的、带有随机性的、重复多轮的分组密码
  - S盒Substitution box（代换）：明文的每一个分组都应产生惟一的一个密文分组，我们把这样的变换称为可逆的，称明文分组到密文分组的可逆变换为代换。
  - P盒Permutation box（置换）：以指定的规则改变比特的排列顺序。
  - 轮密钥（subkey，子密钥）：为每一轮添加的随机数
  - 加密过程：
    1. 轮密钥异或
    2. 代换 (Substitution)
    3. 置换 (Permutation)
    4. 用新的轮密钥开始下一轮加密

- **SPN结构(代换置换网络 Substitution-permutation networks)**



输入16bit的明文和16bit的密钥，输出16bit的密文。分为4个块，每个块中, 分别有4bit明文、密文。共有4轮加密。

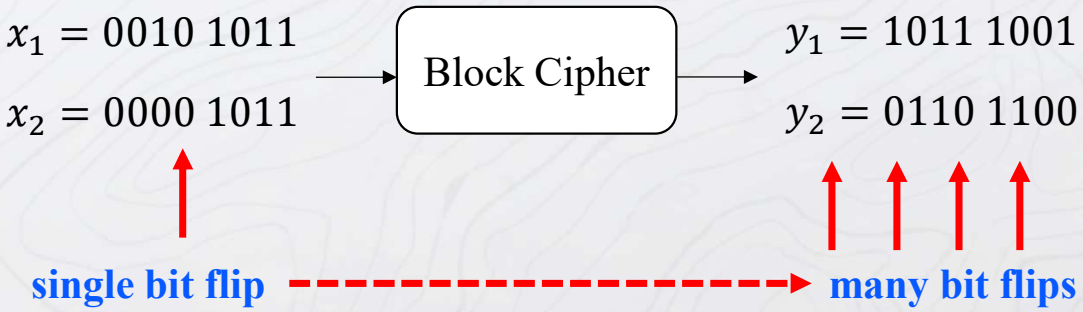
1. 轮密钥异或：本例中，每一轮密钥都未知，每轮密钥之间的关系也未知。
2. 代换：这是一个4x4的s盒，这里的例子中，所有的s盒都相同。但也可以设计不同的s盒。s盒最重要的属性是要能提供非线性的能力，s盒设计的好坏，决定了加密算法抗线性分析的能力。

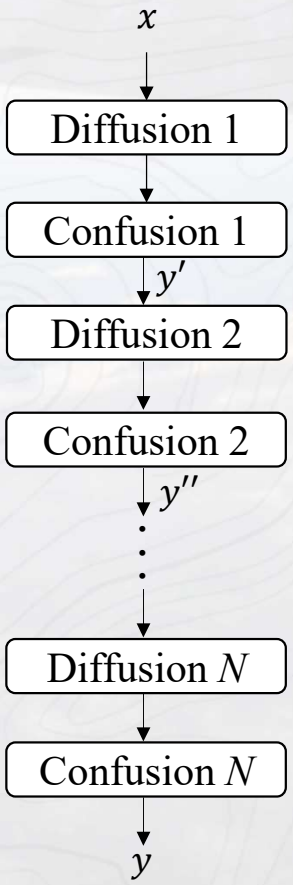
input	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
output	E	4	D	1	2	F	B	8	3	A	6	C	5	9	0	7

3. 置换：通常每轮只有一个 p 盒，每轮 p 盒都相同。p 盒的作用主要使字母失去统计意义。

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

- **SPN结构(代换置换网络 Substitution-permutation networks)**
  - S盒混淆（Confusion）：对称密码中的混淆是通过改变密钥对数据的应用来掩盖输入（明文）和输出（密文）之间的局部相关性，增加密文和密钥间的复杂度。
  - P盒扩散（Diffusion）：扩散是一种加密操作，将一个明文符号的影响扩散到许多密文符号上，目的是隐藏明文的统计特性。





- **Feistel (费斯妥)网络**

1973年霍斯特•费斯妥和Don Coppersmith设计

- 核心思想：轮函数round function——接收两个输入参数，分别是分组数据（原始数据的一半）和子key，然后生成和分组数据同样长度的数据。
- 加密过程：

1. 将明文块拆分为两个等长的块L_0和R_0
2. 对每轮$i = 0,1,\dots,n$计算

$$L_{i+1} = R_i$$
$$R_i + 1 = L_i \oplus F(R_i, K_i)$$

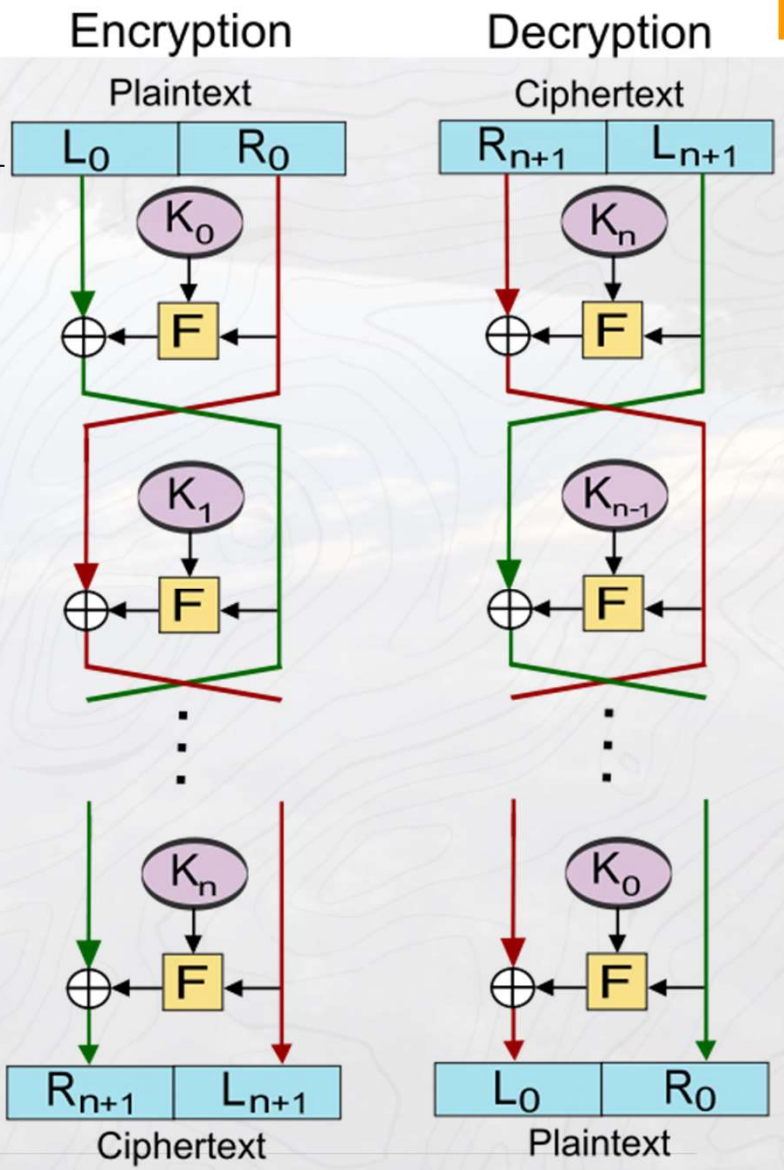
则密文为(R_{n+1}, L_{n+1})

- 解密过程：

1. 将密文块拆分为两个等长的块(R_{n+1}, L_{n+1})
2. 对每轮$i = n, n-2, \dots, 0$ 计算

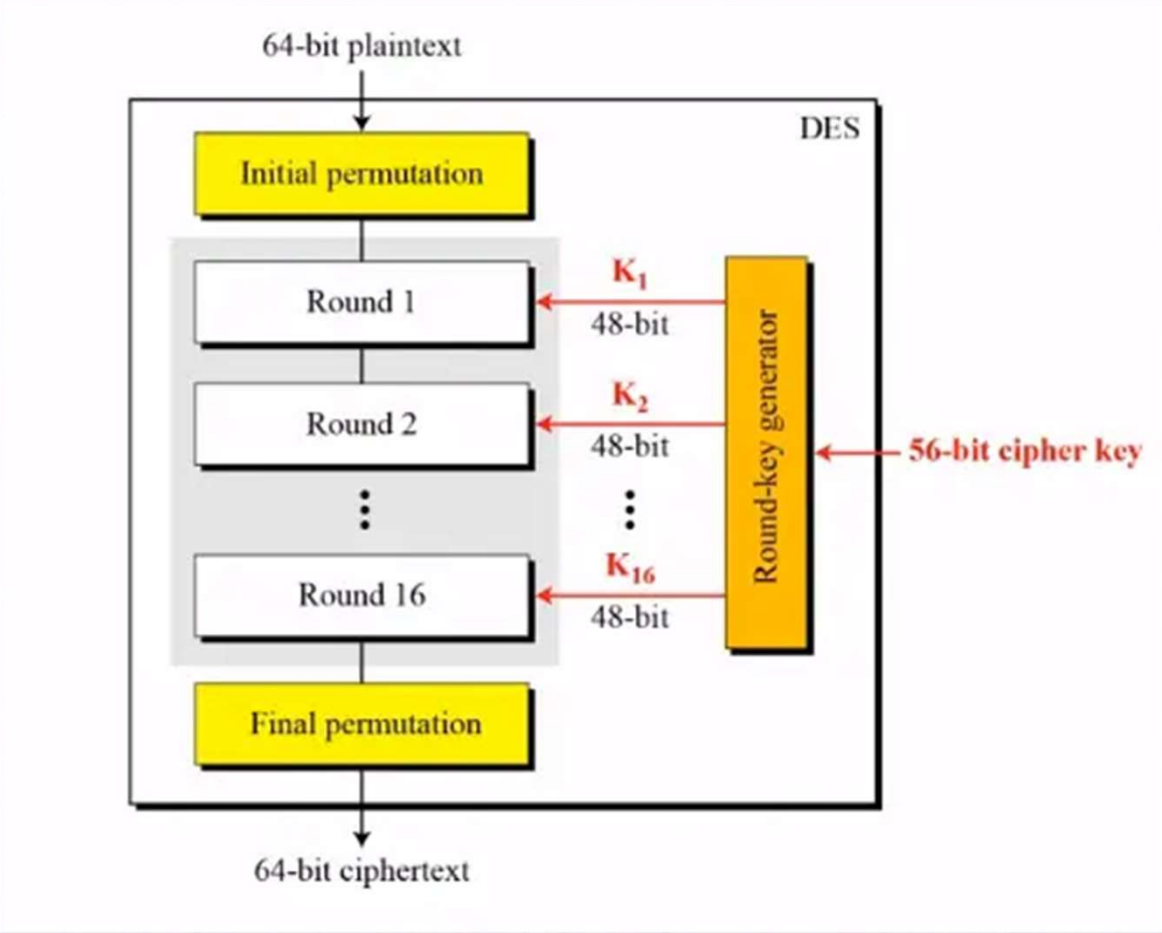
$$R_i = L_{i+1}$$
$$L_i = R_{i+1} \oplus F(L_{i+1}, K_i)$$

则明文为(L_0, R_0)



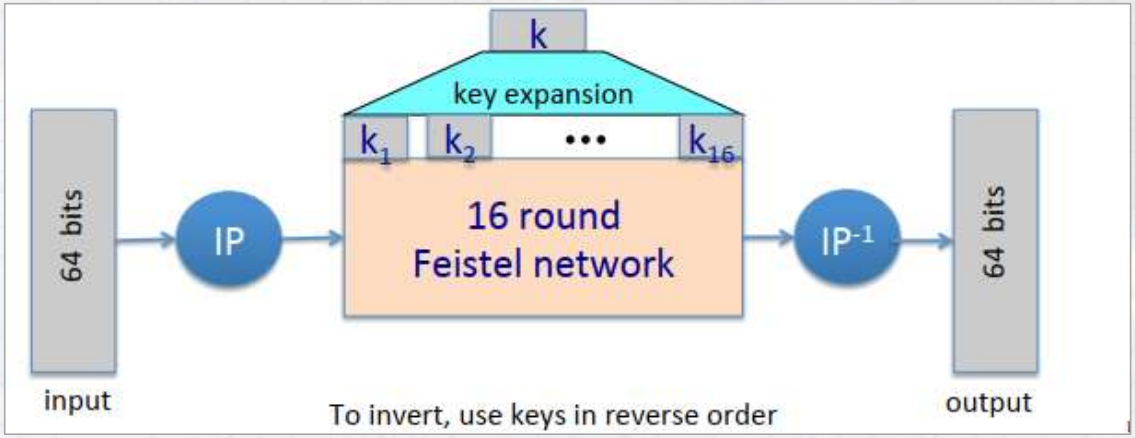
- **Feistel (费斯妥)网络**
  - 轮函数：实现对数据混淆和加密，包括代换、置换和轮密钥异或
  - 关键参数：
    1. 分组大小：分组越大则安全性越高，但加密速度就越慢；
    2. 密钥大小：密钥越长则安全性越高，但加密速度就越慢；
    3. 轮数：单轮结构远不足以保证安全性，但多轮结构可提供足够的安全性，典型地，轮数取为16；
    4. 子密钥产生算法：该算法的复杂性越大，则密码分析的困难性就越大；
    5. 轮函数：轮函数的复杂性越大，密码分析的困难性也越大。
  - 优势：不要求轮函数可逆，能节省时间和空间
  - 劣势：由于轮函数只作用于一半比特，采用暴力破解需要尝试的比特更少

- **DES(Data Encryption Standard 数据加密标准)**
  - 1977年被美国联邦政府的国家标准局（NIST）确定为联邦资料处理标准（FIPS），并授权在非密级政府通信中使用，随后该算法在国际上广泛流传开来。
  - 使用 56 位密钥对 64 位长分组进行加密。
  - 密钥空间较小，因此自 90 年代末以来可以进行穷举搜索攻击。



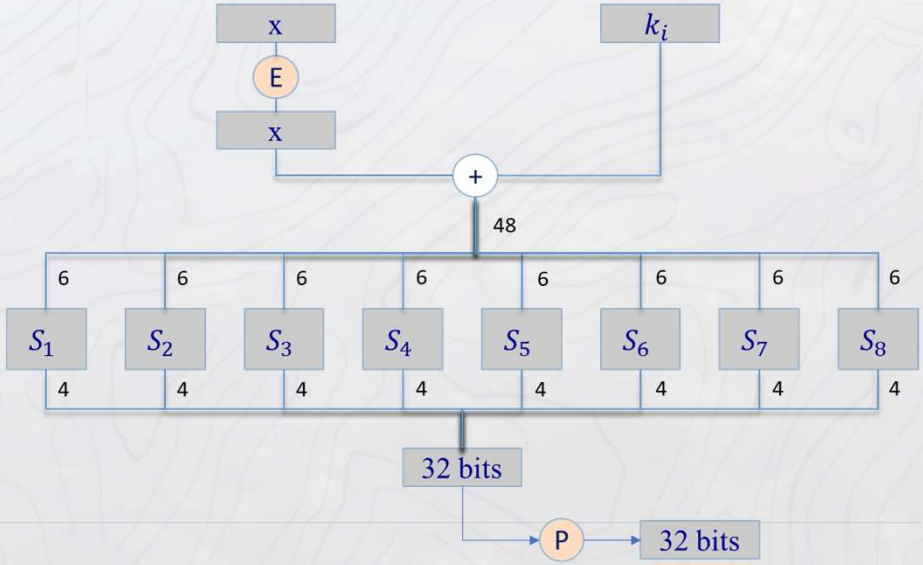
- **DES(Data Encryption Standard 数据加密标准)**

  - 初始置换（IP）：把输入的64位数据块按位重新组合，并把输出分为L0、R0两部分，每部分各长32位
  - 16轮Feistel结构
  - 逆初始置换（IP^{-1}）：把输出的64位密文再次打乱顺序



Initial Permutation							
58	50	42	34	26	18	10	02
60	52	44	36	28	20	12	04
62	54	46	38	30	22	14	06
64	56	48	40	32	24	16	08
57	49	41	33	25	17	09	01
59	51	43	35	27	19	11	03
61	53	45	37	29	21	13	05
63	55	47	39	31	23	15	07

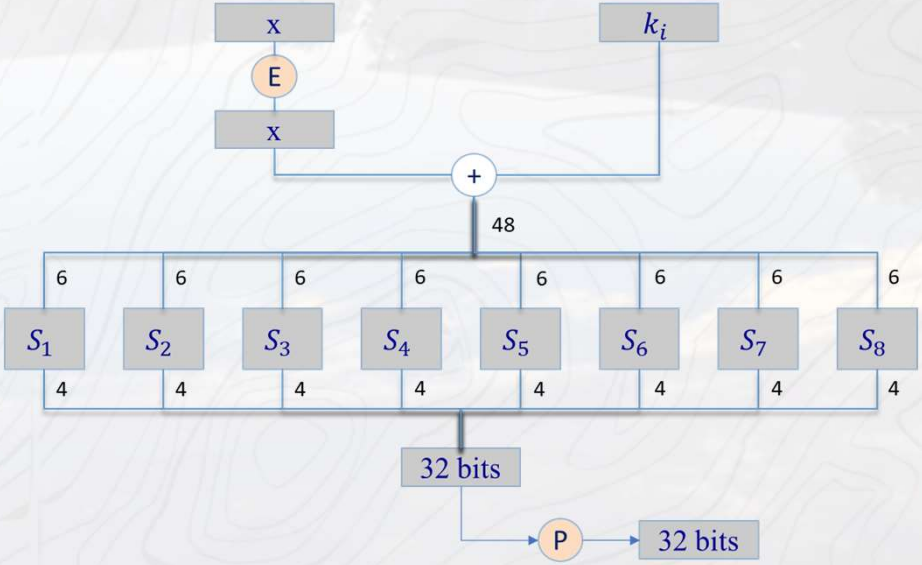
Final Permutation							
40	08	48	16	56	24	64	32
39	07	47	15	55	23	63	31
38	06	46	14	54	22	62	30
37	05	45	13	53	21	61	29
36	04	44	12	52	20	60	28
35	03	43	11	51	19	59	27
34	02	42	10	50	18	58	26
33	01	41	09	49	17	57	25



- **DES(Data Encryption Standard 数据加密标准)**

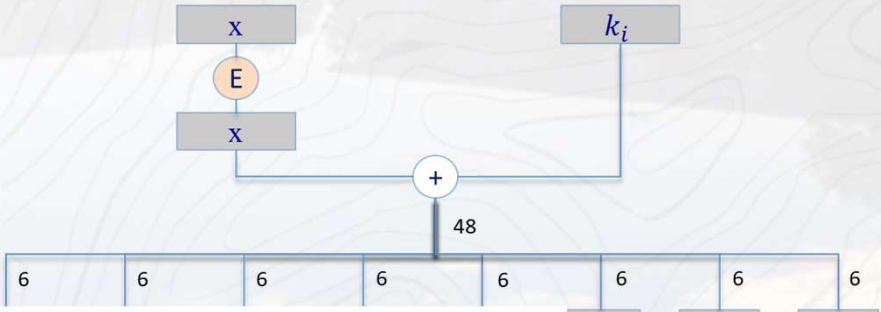
1. E盒：扩展置换，将32位输入扩展为48位

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



- **DES(Data Encryption Standard 数据加密标准)**

1. E盒：扩展置换，将32位输入扩展为48位
2. 扩展后的48位与密钥异或，得到48位密文
3. 48位密文分为8组，分别进入8个S盒，每个S盒输入6位，输出4位，合起来输出32位



S_1																
	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

S_2																
	15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
	3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
	0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
	13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9

S_3																
	10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
	13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1
	13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7
	1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12

S_4																
	7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15
	13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9
	10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4
	3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14

S_5																
	2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
	14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6
	4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14
	11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3

S_6																
	12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11
	10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8
	9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6
	4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13

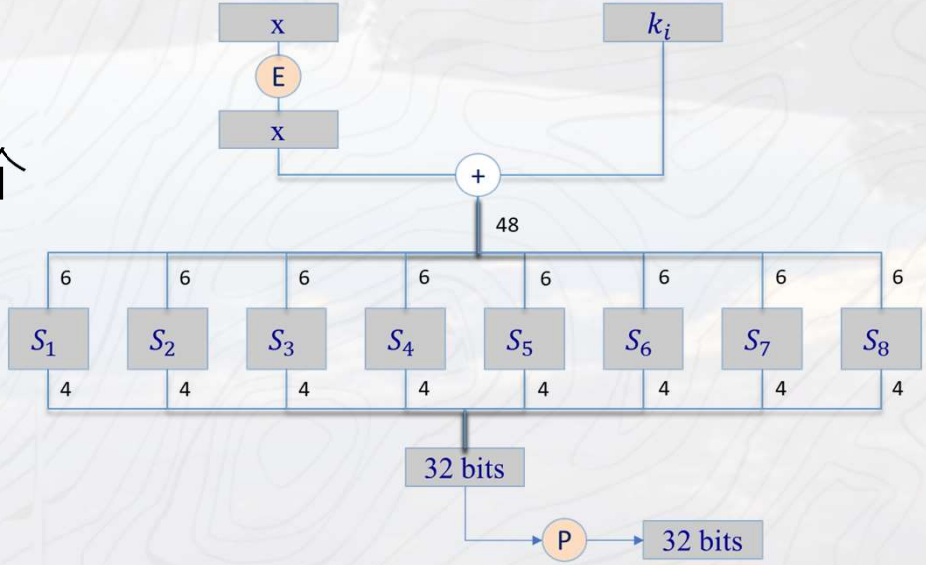
S_7																
	4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1
	13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6
	1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2
	6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12

S_8																
	13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7
	1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2
	7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8
	2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11

- **DES(Data Encryption Standard 数据加密标准)**

1. E盒：扩展置换，将32位输入扩展为48位
2. 扩展后的48位与密钥异或，得到48位密文
3. 48位密文分为8组，分别进入8个S盒，每个S盒输入6位，输出4位，合起来输出32位

对于输入的6位，第1和第6位组成二进制的行号，第2到第5位组成二进制的列号，由此确定的S盒的位置上的10进制数的4位二进制数即为输出。举例来说，输入101001，行号11，列号0100，即第3行第4列（行列从0开始数）。如果是S2的话，对应位置为3，输出即为0011。

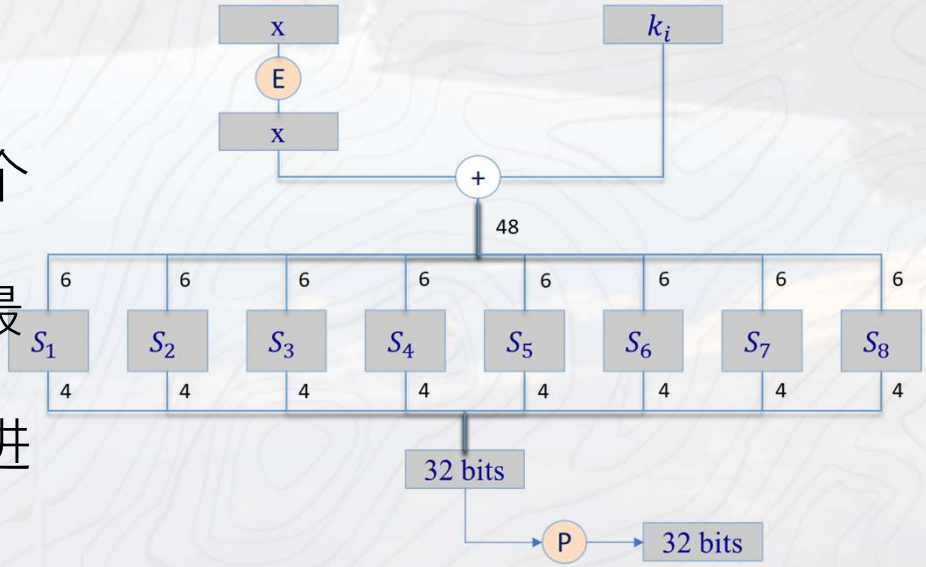


S_2

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

- **DES(Data Encryption Standard 数据加密标准)**

1. E盒：扩展置换，将32位输入扩展为48位
2. 扩展后的48位与密钥异或，得到48位密文
3. 48位密文分为8组，分别进入8个S盒，每个S盒输入6位，输出4位，合起来输出32位
4. P盒：置换得到32位输出，作为轮函数的最终输出
5. 轮函数的32位输出还要与左侧的输入L_{i-1}进行异或，从而得到了右侧输出R_i
6. 开始下一轮Feistel结构



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

- **DES(Data Encryption Standard 数据加密标准)**

密钥产生算法：

1. DES的密钥实际有64位，在输入密钥后，对其1-64位按顺序编号，按照下表所示按顺序放置。
2. 放置后，去掉每行的第8位，形成一个56位的密钥（前文提到的56位）。然后让这56位密钥在置换选择1(PC-1)的作用下置换。
3. 置换后，将获得的56位分成各为28位的两部分C_0, D_0。在每轮迭代中，分别对这两部分进行循环左移（如12345循环左移1位为23451）一位或两位，移位后的值作为下一轮的输入。具体需要移动的位数取决于当前迭代第几轮。

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

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

迭代轮数	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

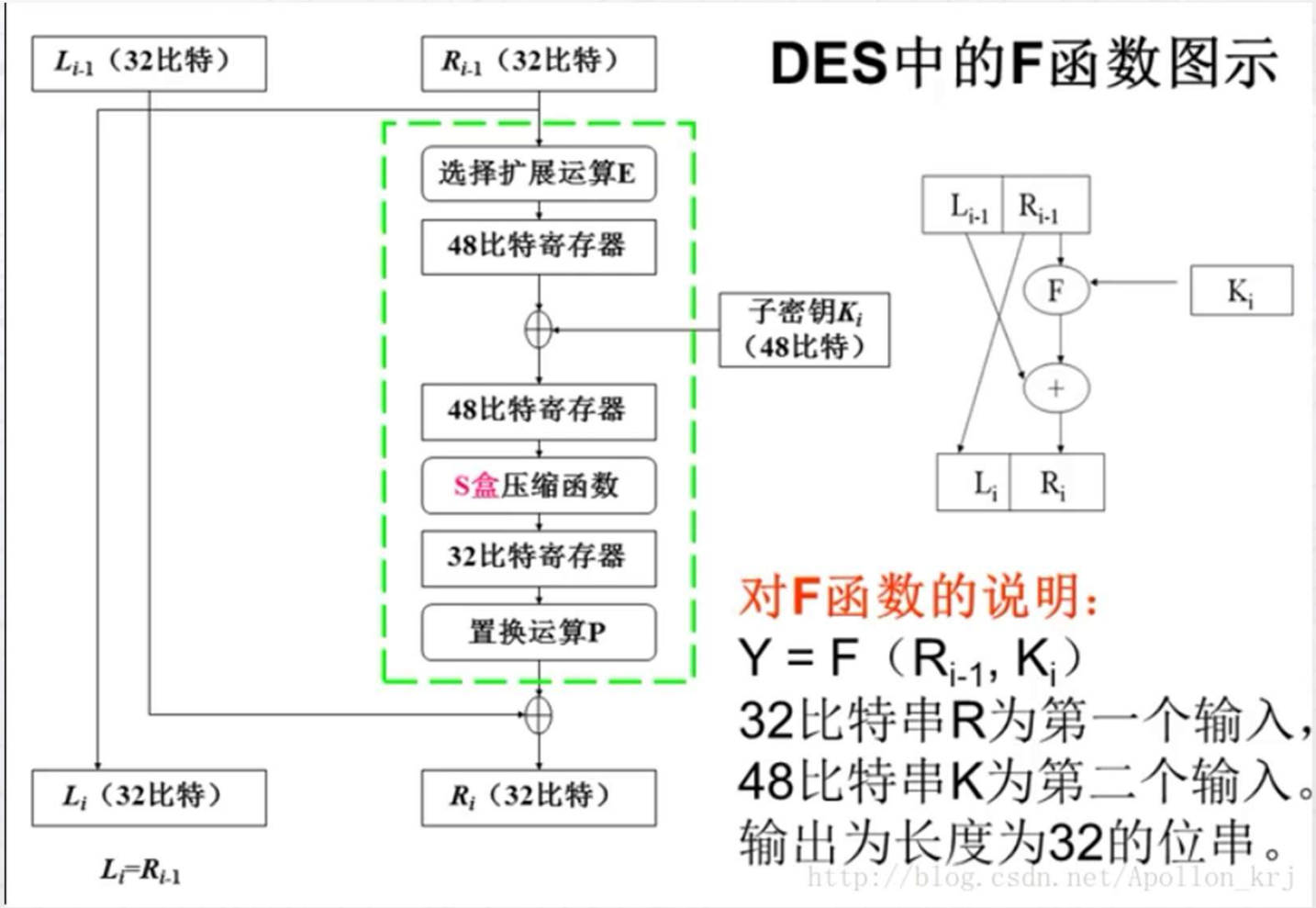
- **DES(Data Encryption Standard 数据加密标准)**

密钥产生算法：

1. DES的密钥实际有64位，在输入密钥后，对其1-64位按顺序编号，按照下表所示按顺序放置。
2. 放置后，去掉每行的第8位，形成一个56位的密钥（前文提到的56位）。然后让这56位密钥在置换选择1(PC-1)的作用下置换。
3. 置换后，将获得的56位分成各为28位的两部分C_0, D_0。在每轮迭代中，分别对这两部分进行循环左移一位或两位，移位后的值作为下一轮的输入。具体需要移动的位数取决于当前迭代第几轮。
4. 然后将移位后获得的2个28位合二为一，对这56位进行置换选择2(PC-2)，得到该轮48位的轮密钥。

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

- **DES(Data Encryption Standard 数据加密标准)**



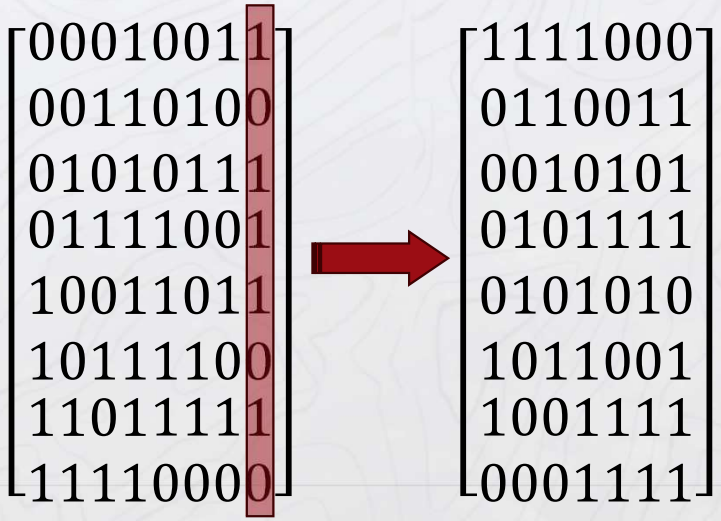
- **DES(Data Encryption Standard 数据加密标准)**

64位明文是0123456789ABCDEF，选取DES密钥133457799BBCDFF1
将明文改写为二进制：0000 0001 0010 0011 0100 0101 0110 0111 1000 1001 1010 1011 1100 1101 1110 1111
将密钥改写为二进制：0001 0011 0011 0100 0101 0111 0111 1001 1001 1011 1011 1100 1101 1111 1111 0001

➢ 子密钥生成阶段：

Step1：密钥按顺序排好，并去掉第8、16、24、32、40、48、56、64位，然后按照给定的置换表进行置换

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



置换后的密钥：
1111000 0110011 0010101 0101111 0101010 1011001 1001111 0001111

- **DES(Data Encryption Standard 数据加密标准)**

64位明文是0123456789ABCDEF，选取DES密钥133457799BBCDFF1

➢ 子密钥生成阶段：

Step2：密钥分为两部分，并按照轮数进行移位

置换后的密钥：
1111000 0110011
0010101 0101111
0101010 1011001
1001111 0001111

划分两部分：
$C_0 = 1111000\ 0110011\ 0010101\ 0101111$
$D_0 = 0101010\ 1011001\ 1001111\ 0001111$

第一轮移动：
$C_1 = 1110000\ 1100110\ 0101010\ 1011111$
$D_1 = 1010101\ 0110011\ 0011110\ 0011110$

迭代轮数	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

- **DES(Data Encryption Standard 数据加密标准)**

64位明文是0123456789ABCDEF，选取DES密钥133457799BBCDFF1

➢ 子密钥生成阶段：

Step2：密钥分为两部分，并按照轮数进行移位

置换后的密钥：
1111000 0110011
0010101 0101111
0101010 1011001
1001111 0001111

第一轮移动：

$C_1 = 1110000\ 1100110\ 0101010\ 1011111$

$D_1 = 1010101\ 0110011\ 0011110\ 0011110$

第二轮移动：

$C_2 = 1100000\ 1100110\ 0101010\ 1011111\ 1$

$D_2 = 0101010\ 0110011\ 0011110\ 0111101$

迭代轮数	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

- **DES(Data Encryption Standard 数据加密标准)**

64位明文是0123456789ABCDEF，选取DES密钥133457799BBCDFF1

➢ 子密钥生成阶段：

Step2：密钥分为两部分，并按照轮数进行移位

置换后的密钥：
1111000 0110011
0010101 0101111
0101010 1011001
1001111 0001111

第二轮移动：
C_2 = 11 0000 1100110 0101010 10111111
D_2 = 01 0101 0110011 0011110 00111101

第三轮移动：
C_3 = 0000 1100110 0101010 10111111 11
D_3 = 0101 0110011 0011110 00111101 01

迭代轮数	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

- **DES(Data Encryption Standard 数据加密标准)**

64位明文是0123456789ABCDEF，选取DES密钥133457799BBCDFF1

➢ 子密钥生成阶段：

Step2：密钥分为两部分，并按照轮数进行移位

置换后的密钥：
1111000 0110011
0010101 0101111
0101010 1011001
1001111 0001111

第十六轮移动：
$C_{16} = 1111000011001100101010101111$
$D_{16} = 0101010101100110011110001111$

迭代轮数	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

- **DES(Data Encryption Standard 数据加密标准)**

64位明文是0123456789ABCDEF，选取DES密钥133457799BBCDFF1

➢ 子密钥生成阶段：

Step3：针对每一轮的子密钥，将两部分合并起来，再次根据置换表进行置换

第一轮移动：

C_1 = 1110000 1100110 0101010 1011111

D_1 = 1010101 0110011 0011110 0011110

置换后K_1

000110 110000 001011 101111

111111 000111 000001 110010

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

- **DES(Data Encryption Standard 数据加密标准)**

64位明文是0123456789ABCDEF，选取DES密钥133457799BBCDFF1

➢ 子密钥生成阶段：

Step3：针对每一轮的子密钥，将两部分合并起来，再次根据置换表进行置换

第二轮移动：

C_2 = 110000 1100110 0101010 10111111

D_2 = 010101 0110011 0011110 00111101

置换后K_2

011110 011010 111011 011001

110110 111100 100111 100101

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

第十六轮移动：

C_{16} = 1111000011001100101010101111

D_{16} = 0101010101100110011110001111

置换后K_{16}

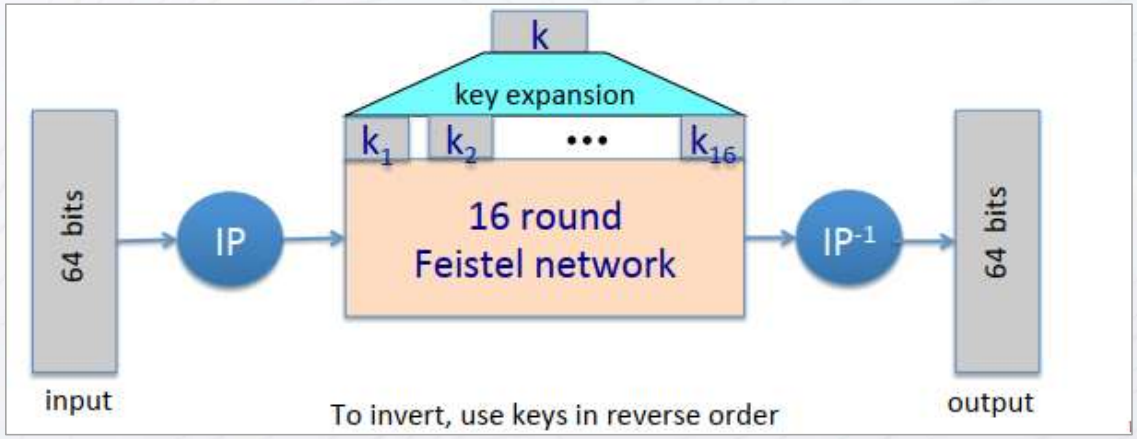
110010 110011 110110 001011

000011 100001 011111 110101

- **DES(Data Encryption Standard 数据加密标准)**

64位明文是0123456789ABCDEF， 选取DES密钥133457799BBCDFF1

➢ 子密钥生成阶段:



置换后K_1
000110 110000 001011 101111
111111 000111 000001 110010

置换后K_2
011110 011010 111011 011001
110110 111100 100111 100101

●●●●●●

置换后K_{16}
110010 110011 110110 001011
000011 100001 011111 110101

- **DES(Data Encryption Standard 数据加密标准)**

64位明文是0123456789ABCDEF，选取DES密钥133457799BBCDFF1
将明文改写为二进制：0000 0001 0010 0011 0100 0101 0110 0111 1000 1001 1010 1011 1100 1101 1110 1111

➢ 明文初始变换：

Initial Permutation							
58	50	42	34	26	18	10	02
60	52	44	36	28	20	12	04
62	54	46	38	30	22	14	06
64	56	48	40	32	24	16	08
57	49	41	33	25	17	09	01
59	51	43	35	27	19	11	03
61	53	45	37	29	21	13	05
63	55	47	39	31	23	15	07

0000 0001 0010 0011
0100 0101 0110 0111
1000 1001 1010 1011
1100 1101 1110 1111

1100 1100 0000 0000
1100 1100 1111 1111
1111 0000 1010 1010
1111 0000 1010 1010

- **DES(Data Encryption Standard 数据加密标准)**

64位明文是0123456789ABCDEF，选取DES密钥133457799BBCDFF1
将明文改写为二进制：0000 0001 0010 0011 0100 0101 0110 0111 1000 1001 1010 1011 1100 1101 1110 1111

➢ 划分L0和R0：

1100 1100 0000 0000
1100 1100 1111 1111

L0 = 1100 1100 0000 0000 1100 1100 1111 1111

1111 0000 1010 1010
1111 0000 1010 1010

R0 = 1111 0000 1010 1010 1111 0000 1010 1010

- **DES(Data Encryption Standard 数据加密标准)**

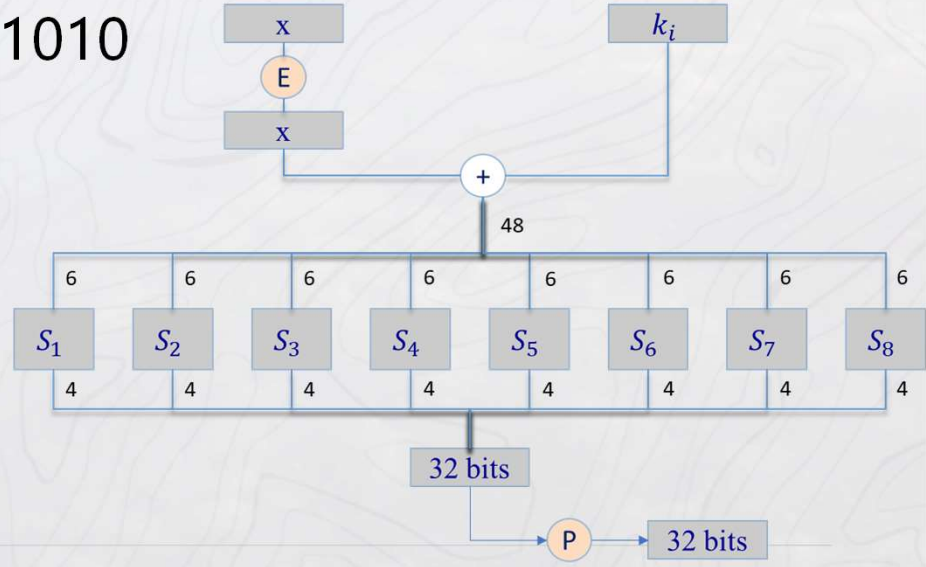
64位明文是0123456789ABCDEF，选取DES密钥133457799BBCDFF1
将明文改写为二进制：0000 0001 0010 0011 0100 0101 0110 0111 1000 1001 1010 1011 1100 1101 1110 1111

➢ 第一轮变换：L0不变，R0进入F函数，开始第一轮变换

L0 = 1100 1100 0000 0000 1100 1100 1111 1111

R0 = 1111 0000 1010 1010 1111 0000 1010 1010

➢ F函数第一步：E变换



- **DES(Data Encryption Standard 数据加密标准)**

64位明文是0123456789ABCDEF，选取DES密钥133457799BBCDFF1
将明文改写为二进制：0000 0001 0010 0011 0100 0101 0110 0111 1000 1001 1010 1011 1100 1101 1110 1111

➢ 第一轮变换：L0不变，R0进入F函数，开始第一轮变换

R0 = 1111 0000 1010 1010 1111 0000 1010 1010

➢ F函数第一步：E变换

E(R0) = 011110 100001 010101 010101 011110 100001 010101 010101

（注意输入的每4位一个分组被拓展为输出的每6位一个分组。）

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

- **DES(Data Encryption Standard 数据加密标准)**

64位明文是0123456789ABCDEF，选取DES密钥133457799BBCDFF1
将明文改写为二进制：0000 0001 0010 0011 0100 0101 0110 0111 1000 1001 1010 1011 1100 1101 1110 1111

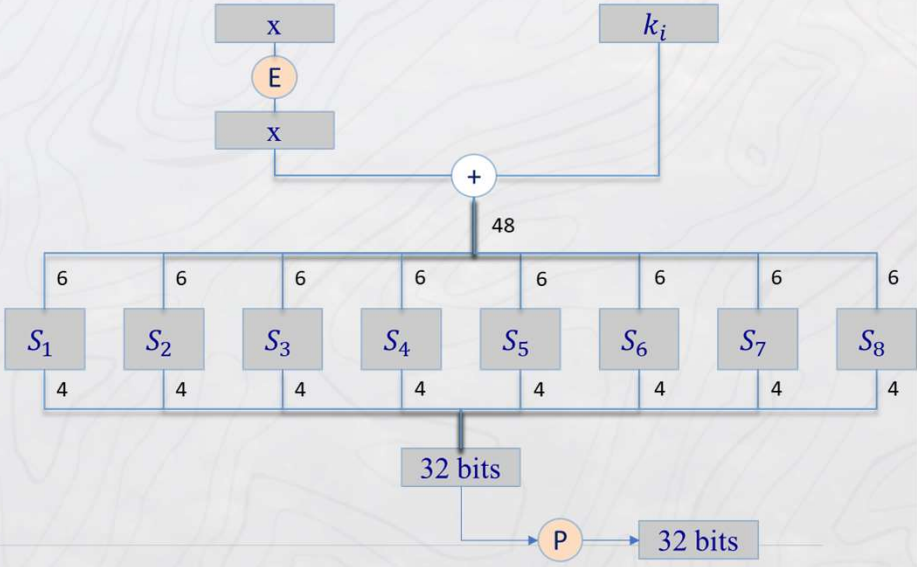
➢ 第一轮变换：L0不变，R0进入F函数，开始第一轮变换

E(R0) = 011110 100001 010101 010101 011110 100001 010101 010101

➢ F函数第二步：轮密钥异或

置换后K_1
000110 110000 001011 101111
111111 000111 000001 110010

K1+E(R0) =
011000 010001 011110 111010
100001 100110 010100 100111



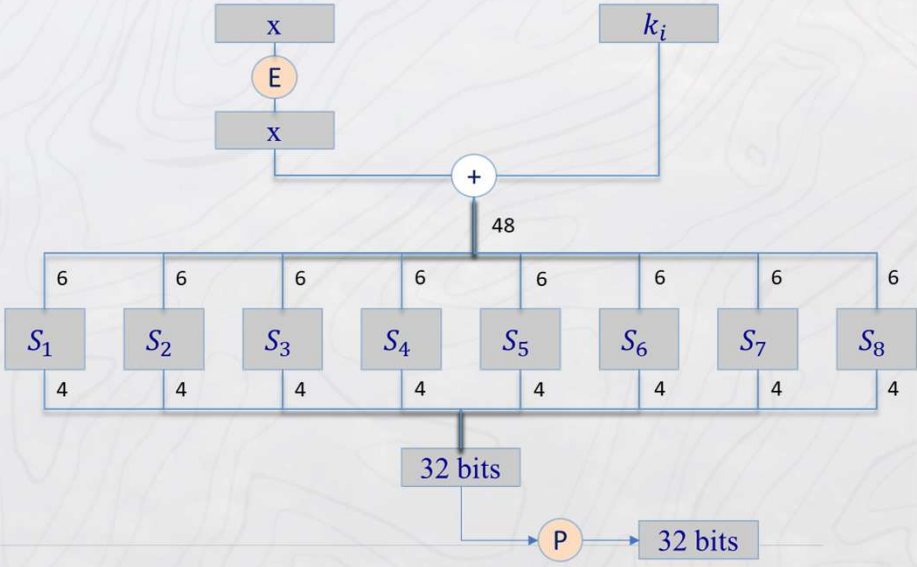
- **DES(Data Encryption Standard 数据加密标准)**

64位明文是0123456789ABCDEF，选取DES密钥133457799BBCDFF1
将明文改写为二进制：0000 0001 0010 0011 0100 0101 0110 0111 1000 1001 1010 1011 1100 1101 1110 1111

➢ 第一轮变换：L0不变，R0进入F函数，开始第一轮变换

K1+E(R0)=011000 010001 011110 111010 100001 100110 010100 100111

➢ F函数第三步：S盒



- **DES(Data Encryption Standard 数据加密标准)**

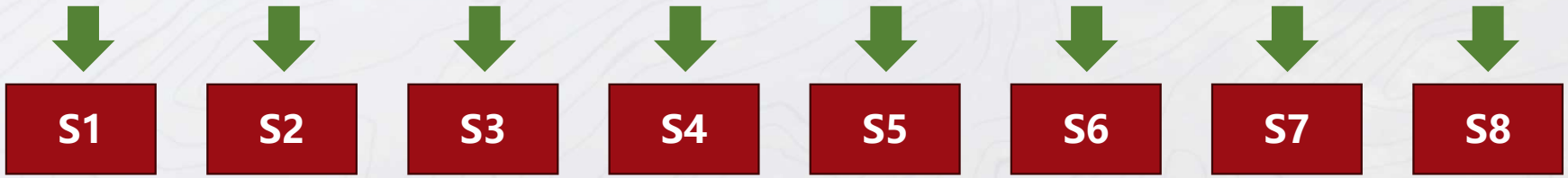
64位明文是0123456789ABCDEF，选取DES密钥133457799BBCDFF1
将明文改写为二进制：0000 0001 0010 0011 0100 0101 0110 0111 1000 1001 1010 1011 1100 1101 1110 1111

- 第一轮变换：L0不变，R0进入F函数，开始第一轮变换

K1+E(R0)=011000 010001 011110 111010 100001 100110 010100 100111

- F函数第三步：S盒

011000 010001 011110 111010 100001 100110 010100 100111



- **DES(Data Encryption Standard 数据加密标准)**

64位明文是0123456789ABCDEF，选取DES密钥133457799BBCDFF1
将明文改写为二进制：0000 0001 0010 0011 0100 0101 0110 0111 1000 1001 1010 1011 1100 1101 1110 1111

➢ 第一轮变换：L0不变，R0进入F函数，开始第一轮变换

K1+E(R0)=011000 010001 011110 111010 100001 100110 010100 100111

➢ F函数第三步：S盒

011000

S1

S_1																
	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

第0行第12列：**5→0101**

- **DES(Data Encryption Standard 数据加密标准)**

64位明文是0123456789ABCDEF，选取DES密钥133457799BBCDFF1
将明文改写为二进制：0000 0001 0010 0011 0100 0101 0110 0111 1000 1001 1010 1011 1100 1101 1110 1111

➢ 第一轮变换：L0不变，R0进入F函数，开始第一轮变换

K1+E(R0)=011000 010001 011110 111010 100001 100110 010100 100111

➢ F函数第三步：S盒

010001

S1

S_2

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

第1行第8列：**12→1100**

- **DES(Data Encryption Standard 数据加密标准)**

64位明文是0123456789ABCDEF，选取DES密钥133457799BBCDFF1
将明文改写为二进制：0000 0001 0010 0011 0100 0101 0110 0111 1000 1001 1010 1011 1100 1101 1110 1111

- 第一轮变换：L0不变，R0进入F函数，开始第一轮变换

K1+E(R0)=011000 010001 011110 111010 100001 100110 010100 100111

- F函数第三步：S盒

011110

S1

S_3															
	10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
	13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1
	13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7
	1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12

第？ 行第？ 列：？ →?

- **DES(Data Encryption Standard 数据加密标准)**

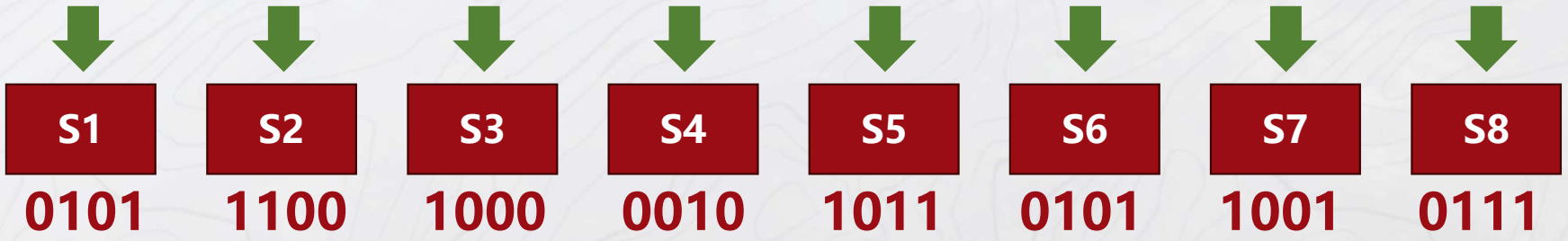
64位明文是0123456789ABCDEF，选取DES密钥133457799BBCDFF1
将明文改写为二进制：0000 0001 0010 0011 0100 0101 0110 0111 1000 1001 1010 1011 1100 1101 1110 1111

- 第一轮变换：L0不变，R0进入F函数，开始第一轮变换

K1+E(R0)=011000 010001 011110 111010 100001 100110 010100 100111

- F函数第三步：S盒

011000 010001 011110 111010 100001 100110 010100 100111



- **DES(Data Encryption Standard 数据加密标准)**

64位明文是0123456789ABCDEF，选取DES密钥133457799BBCDFF1
将明文改写为二进制：0000 0001 0010 0011 0100 0101 0110 0111 1000 1001 1010 1011 1100 1101 1110 1111

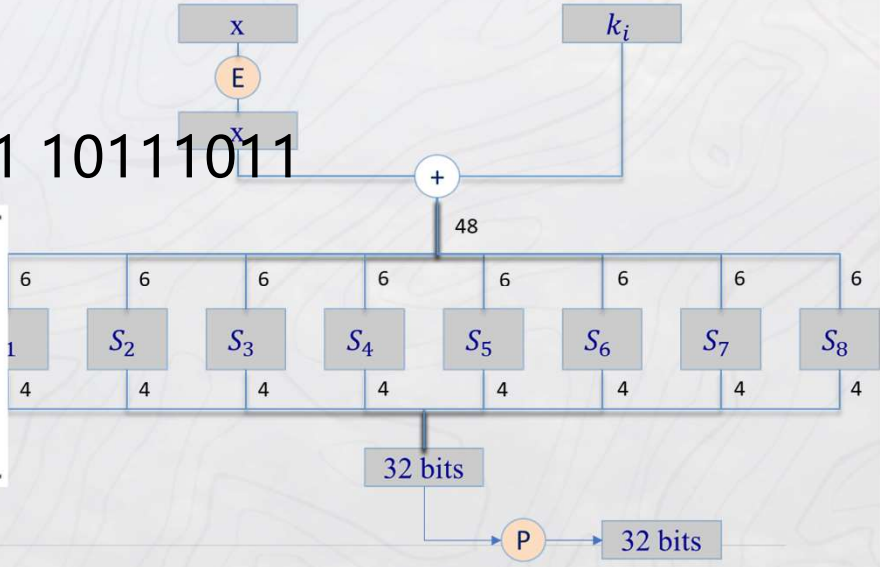
➢ 第一轮变换：L0不变，R0进入F函数，开始第一轮变换

S(K1+E(R0))=0101 1100 1000 0010 1011 0101 1001 0111

➢ F函数第四步：P盒

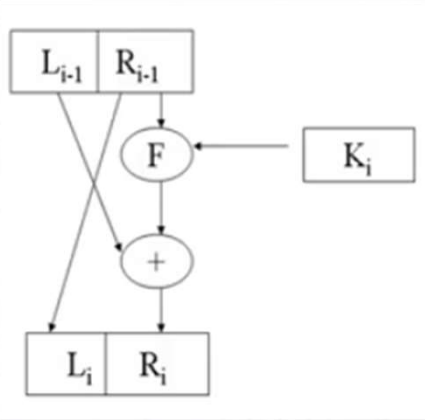
P(S(K1+E(R0)))=00100011 01001010 10101001 10111011

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



- **DES(Data Encryption Standard 数据加密标准)**

64位明文是0123456789ABCDEF，选取DES密钥133457799BBCDFF1



第一轮变换输出0010 0011 0100 1010 1010 1001 1011 1011

L0 = 1100 1100 0000 0000 1100 1100 1111 1111

异或运算：1110 1111 0100 1010 0110 0101 0100 0100

在下一轮中：L1= 1111 0000 1010 1010 1111 0000 1010 1010

R1= 1110 1111 0100 1010 0110 0101 0100 0100

K = 011110 011010 111011 011001 110110 111100 100111 100101

➢ 开始第二轮Feistel结构的变换

- **DES(Data Encryption Standard 数据加密标准)**

64位明文是0123456789ABCDEF，选取DES密钥133457799BBCDFF1

➢ 16轮变换结束，开始最终变换

L16 = 0100 0011 0100 0010 0011 0010 0011 0100

R16 = 0000 1010 0100 1100 1101 1001 1001 0101

调换位置得到：

R16L16 = 00001010 01001100 11011001 10010101 01000011 01000010 00110010 00110100

IP-1 = 10000101 11101000 00010011 01010100 00001111 00001010 10110100 00000101

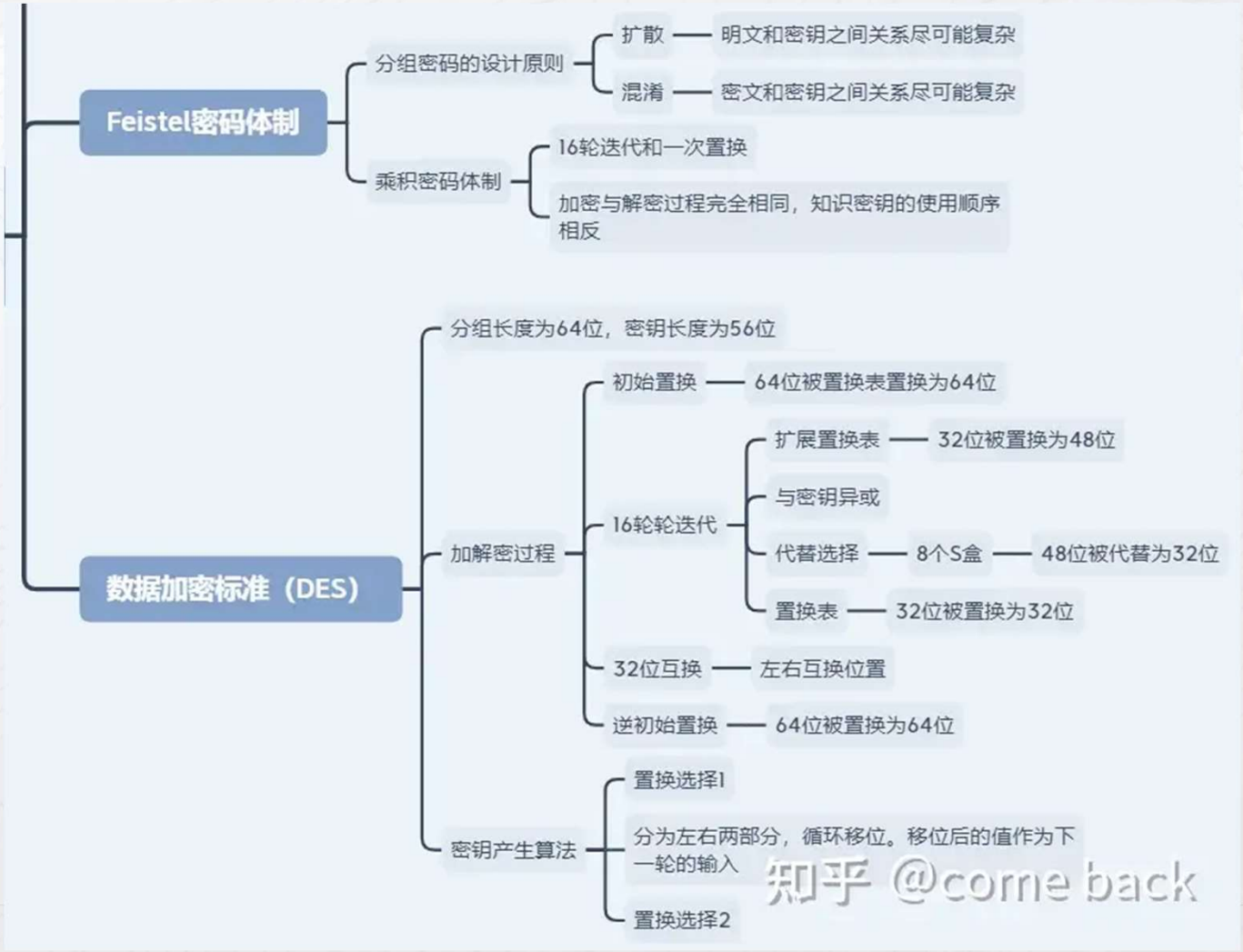
转16进制：85E813540F0AB405

Final Permutation

40	08	48	16	56	24	64	32
39	07	47	15	55	23	63	31
38	06	46	14	54	22	62	30
37	05	45	13	53	21	61	29
36	04	44	12	52	20	60	28
35	03	43	11	51	19	59	27
34	02	42	10	50	18	58	26
33	01	41	09	49	17	57	25

明文M = 0123456789ABCDEF的加密形式C = 85E813540F0AB405

- **DES(Data Encryption Standard 数据加密标准)**



- **DES(Data Encryption Standard 数据加密标准)**

面临问题：密钥空间小，暴力破解风险高

1997年1月28日，美国的RSA数据安全公司在互联网上开展了一项名为"密钥挑战"的竞赛，悬赏一万美元，破解一段用56位密钥加密的DES密文。

msg = "The unknown messages is: XXXX …"

CT = c_1 c_2 c_3 c_4

Goal: find $k \in \{0,1\}^{56}\ s.t.\ DES(k, m_i) = c_i\ for\ i = 1,2,3$

1997: 互联网分段运行程序Internet search -- **3 months**

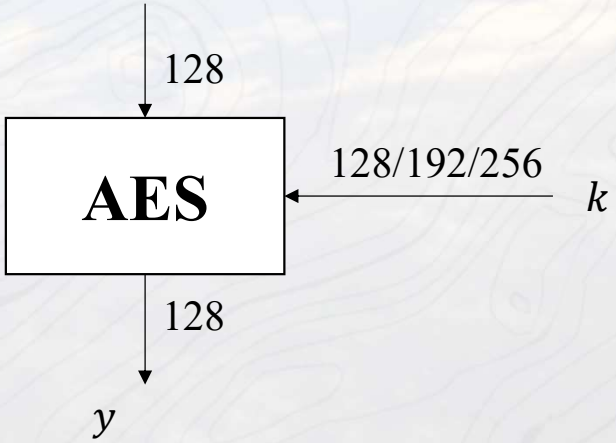
1998: EFF machine (deep crack) –**3days** (250K$)

1999: combined search – **22hours**

2006: COPACOBANA (120 FPGAs) **--7days** (10K$)

⇒ 56-bit ciphers should not be used !! (128-bit key ⇒ 2^{72}days)

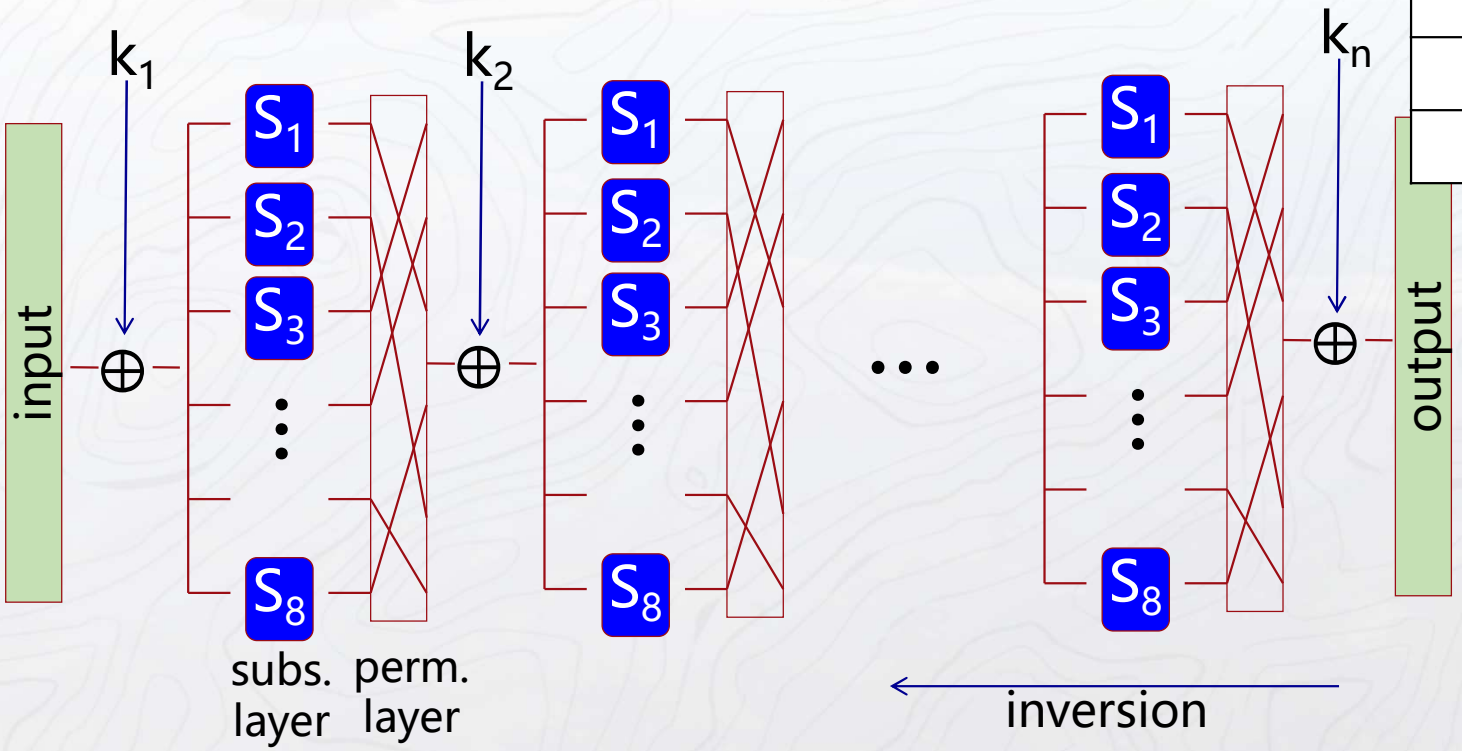
- **AES(Advanced Encryption Standard 高级加密标准)**
  - ❑ AES 是当今使用最广泛的对称密码
  - ❑ AES 的算法是由美国国家标准与技术研究院（NIST）通过多年的筛选过程选出的
  - ❑ 对所有候选 AES 算法的要求是
    - ✓ 块大小为 128 位的分组密码
    - ✓ 支持三种密钥长度：128、192 和 256 位
    - ✓ 相对于其他候选算法和已有的DES算法更安全
    - ✓ 综合考虑软件和硬件的效率
    - ✓ 可抵御已知攻击



- **AES(Advanced Encryption Standard 高级加密标准)**
  - 1997 年 1 月，NIST 宣布需要一种新的分组密码
  - 1998 年 8 月接受 15 种候选算法
  - 1999 年 8 月公布 5 个入围者：
    - ✓ 火星--IBM 公司
    - ✓ RC6 - RSA 实验室
    - ✓ Rijndael - J. Daemen 和 V. Rijmen
    - ✓ 蛇形-Eli Biham 等人
    - ✓ Twofish - B.Schneier 等人
  - 2000 年 10 月，Rijindael 被选为 AES 的名称。
  - 2001 年 11 月，AES 被正式批准为美国联邦标准

- **AES(Advanced Encryption Standard 高级加密标准)**

AES is a SP network (not Feistel)

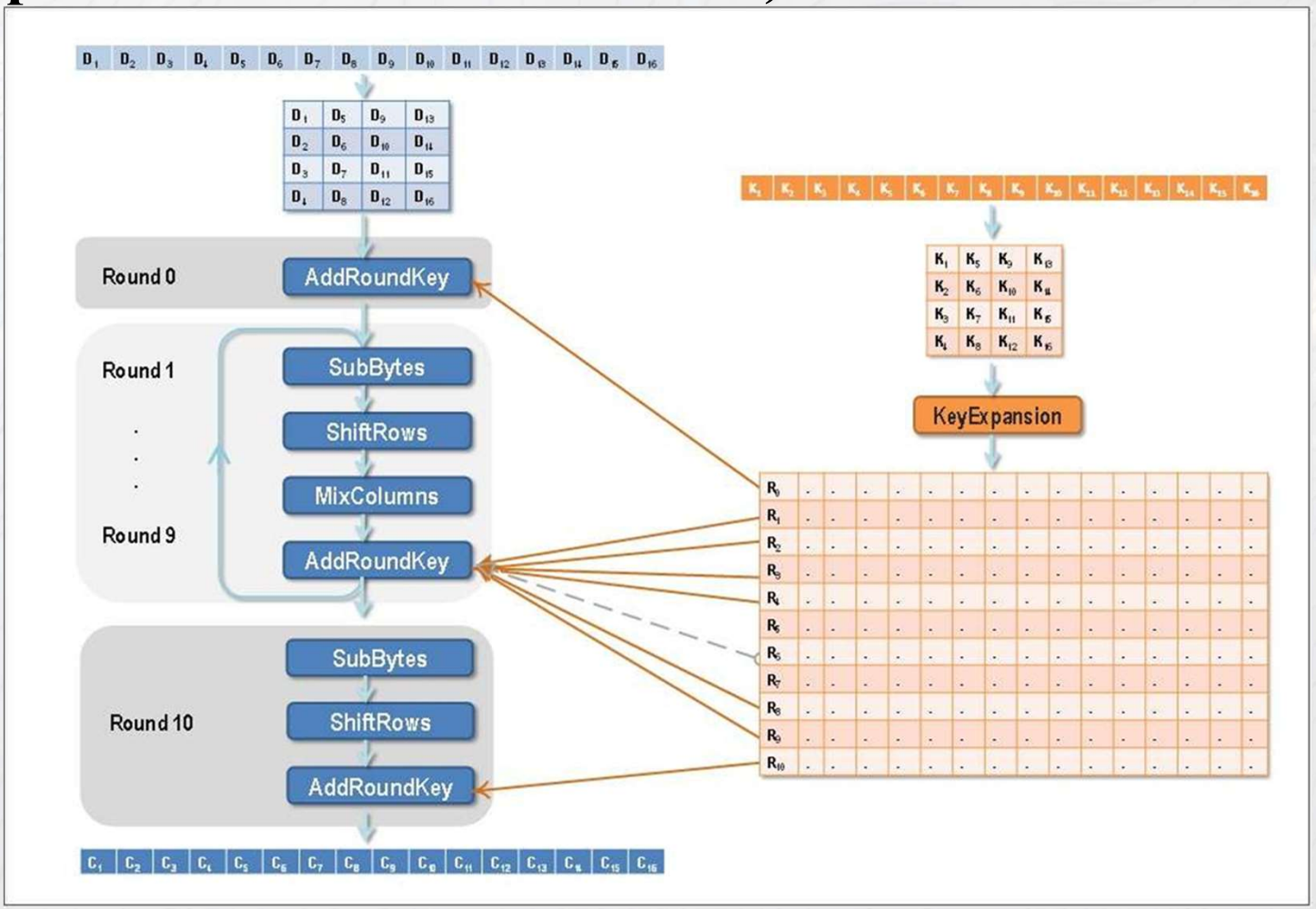


Key length (bits)	Number of rounds
128	10
192	12
256	14

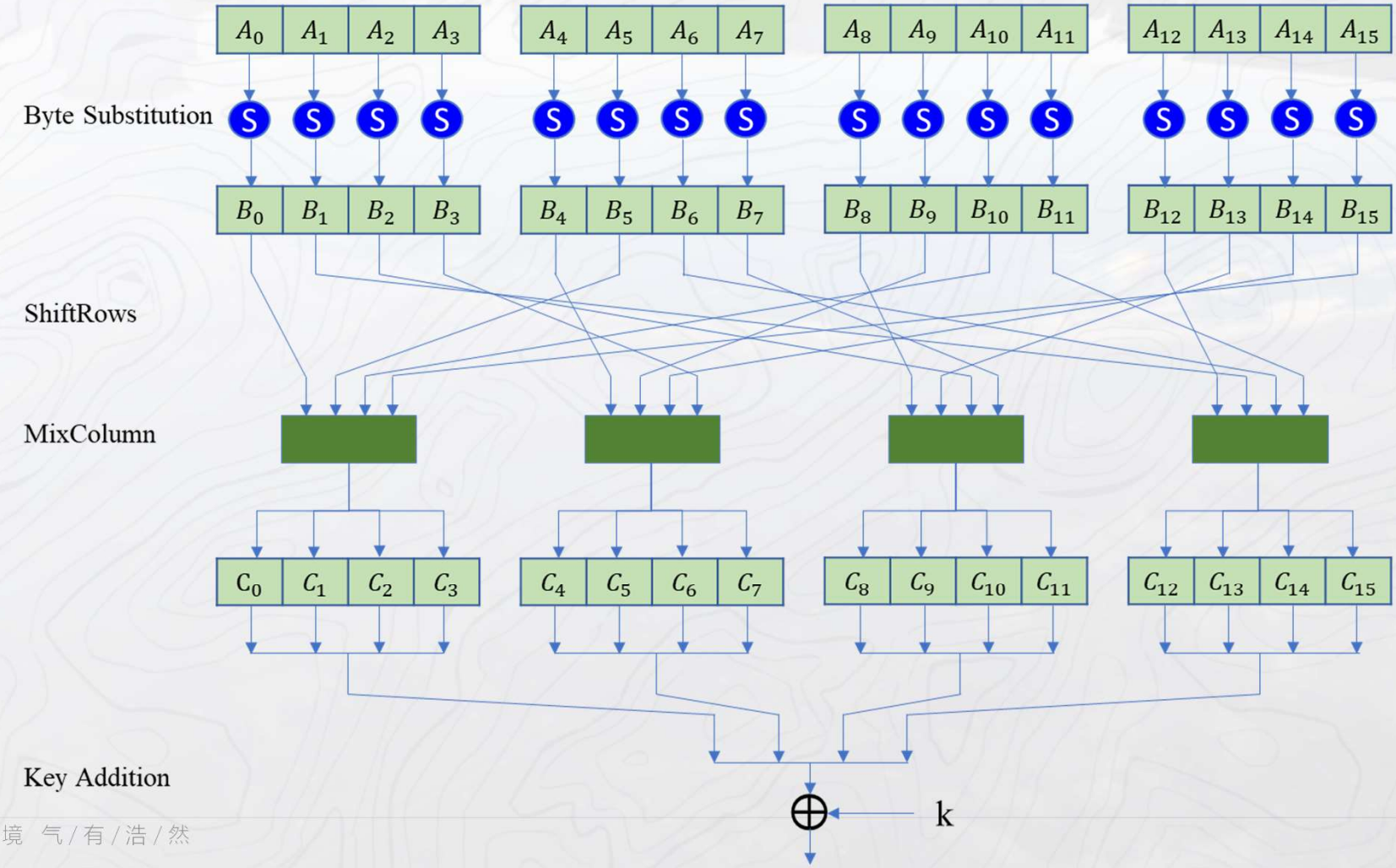
轮数取决于密钥长度

- **AES(Advanced Encryption Standard 高级加密标准)**

1. 字节代换
2. 行位移
3. 列混淆
4. 轮密钥异或



- **AES(Advanced Encryption Standard 高级加密标准)——轮函数**



- **AES(Advanced Encryption Standard 高级加密标准)——轮函数**

1. 字节代换——S盒

❑ 字节替换层由 16 个 S-Boxes 组成，具有以下特性：

S盒是

- 相同的
- AES的唯一非线性元素，即

$ByteSub(A_i) + ByteSub(A_j) \neq ByteSub(A_i + A_j), for\ i,\ j = 0, \ldots, 15$

- 双射，即存在输入和输出字节的一一映射

→S-Box可逆且逆向唯一

❑ 在软件实施过程中，S-Box 通常以查找表的形式实现

- **AES(Advanced Encryption Standard 高级加密标准)——轮函数**

1. 字节代换——S盒

- 字节替换层由 16 个 S-Box
  S盒是
  - 相同的
  - AES的唯一非线性
  $ByteSub(A_i) + Byte$
  - 双射，即存在输入
  →S-Box可逆且逆向唯
- 在软件实施过程中，S-Bo

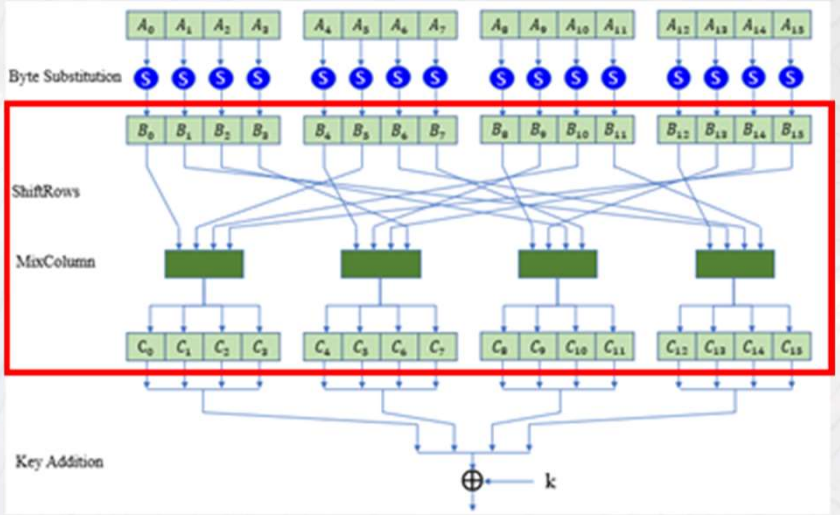
X \ Y	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	00	01	8D	F6	CB	52	7B	D1	E8	4F	29	C0	B0	E1	E5	C7
1	74	B4	AA	4B	99	2B	60	5F	58	3F	FD	CC	FF	40	EE	B2
2	3A	6E	5A	F1	55	4D	A8	C9	C1	0A	98	15	30	44	A2	C2
3	2C	45	92	6C	F3	39	66	42	F2	35	20	6F	77	BB	59	19
4	1D	FE	37	67	2D	31	F5	69	A7	64	AB	13	54	25	E9	09
5	ED	5C	05	CA	4C	24	87	BF	18	3E	22	F0	51	EC	61	17
6	16	5E	AF	D3	49	A6	36	43	F4	47	91	DF	33	93	21	3B
7	79	B7	97	85	10	B5	BA	3C	B6	70	D0	06	A1	FA	81	82
8	83	7E	7F	80	96	73	BE	56	9B	9E	95	D9	F7	02	B9	A4
9	DE	6A	32	6D	D8	8A	84	72	2A	14	9F	88	F9	DC	89	9A
A	FB	7C	2E	C3	8F	B8	65	48	26	C8	12	4A	CE	E7	D2	62
B	0C	E0	1F	EF	11	75	78	71	A5	8E	76	3D	BD	BC	86	57
C	0B	28	2F	A3	DA	D4	E4	0F	A9	27	53	04	1B	FC	AC	E6
D	7A	07	AE	63	C5	DB	E2	EA	94	8B	C4	D5	9D	F8	90	6B
E	B1	0D	D6	EB	C6	0E	CF	AD	08	4E	D7	E3	5D	50	1E	B3
F	5B	23	38	34	68	46	03	8C	DD	9C	7D	A0	CD	1A	41	1C

- **AES(Advanced Encryption Standard 高级加密标准)——轮函数**

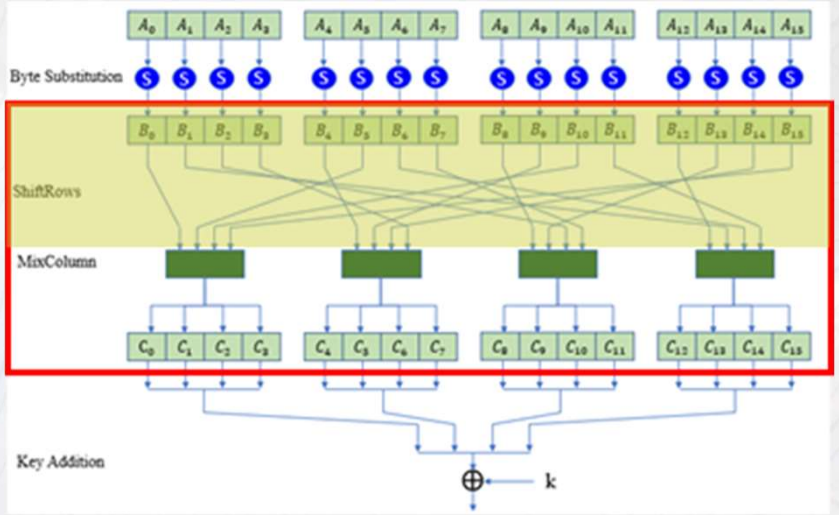
2. 混淆层——P盒

- 行位移：字节级数据置换
- 列混淆：矩阵操作，用于组合（"混合"）四个字节的数据块
- 是线性变换：

$$DIFF(A) + DIFF(B) = DIFF(A + B)$$



- **AES(Advanced Encryption Standard 高级加密标准)——轮函数**



2. 混淆层——P盒

❑ 行位移：字节级数据置换

Input matrix

B_0	B_4	B_8	B_{12}
B_1	B_5	B_9	B_{13}
B_2	B_6	B_{10}	B_{14}
B_3	B_7	B_{11}	B_{15}

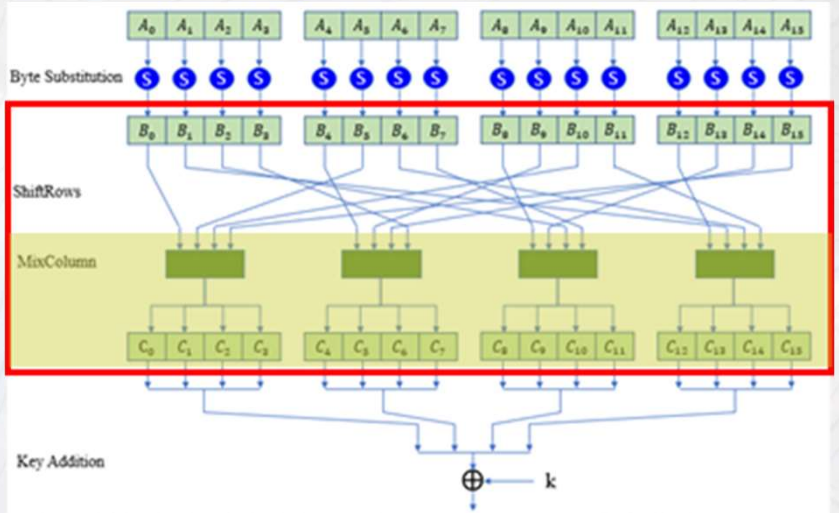
Output matrix

B_0	B_4	B_8	B_{12}	no shift
B_5	B_9	B_{13}	B_1	← one position left shift
B_{10}	B_{14}	B_2	B_6	← two position left shift
B_{15}	B_3	B_7	B_{11}	← three position left shift

- **AES(Advanced Encryption Standard 高级加密标准)——轮函数**

2. 混淆层——P盒

- 列混淆：矩阵操作，用于组合（“混合”）四个字节的数据块，实质是在有限域GF(256) 上的多项式乘法运算。



GF 伽罗华域（**Galois Field**）
域是一种定义了域中元素两种数学运算的代数系统，域由全体元素的加法集合以及非零元素的乘法集合构成。对域中的元素进行加法或乘法运算后的结果仍然是域中的元素。

具有有限个元素的域，称为有限域，即伽罗华域。

这里的加法和乘法并不一定是四则运算中的加和乘，而是指与运算和异或运算

- **AES(Advanced Encryption Standard 高级加密标准)——轮函数**

2. 混淆层——P盒

- 列混淆：矩阵操作，用于组合（“混合”）四个字节的数据块，实质是在有限域GF(256) 上的多项式乘法运算。

$$\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix}$$

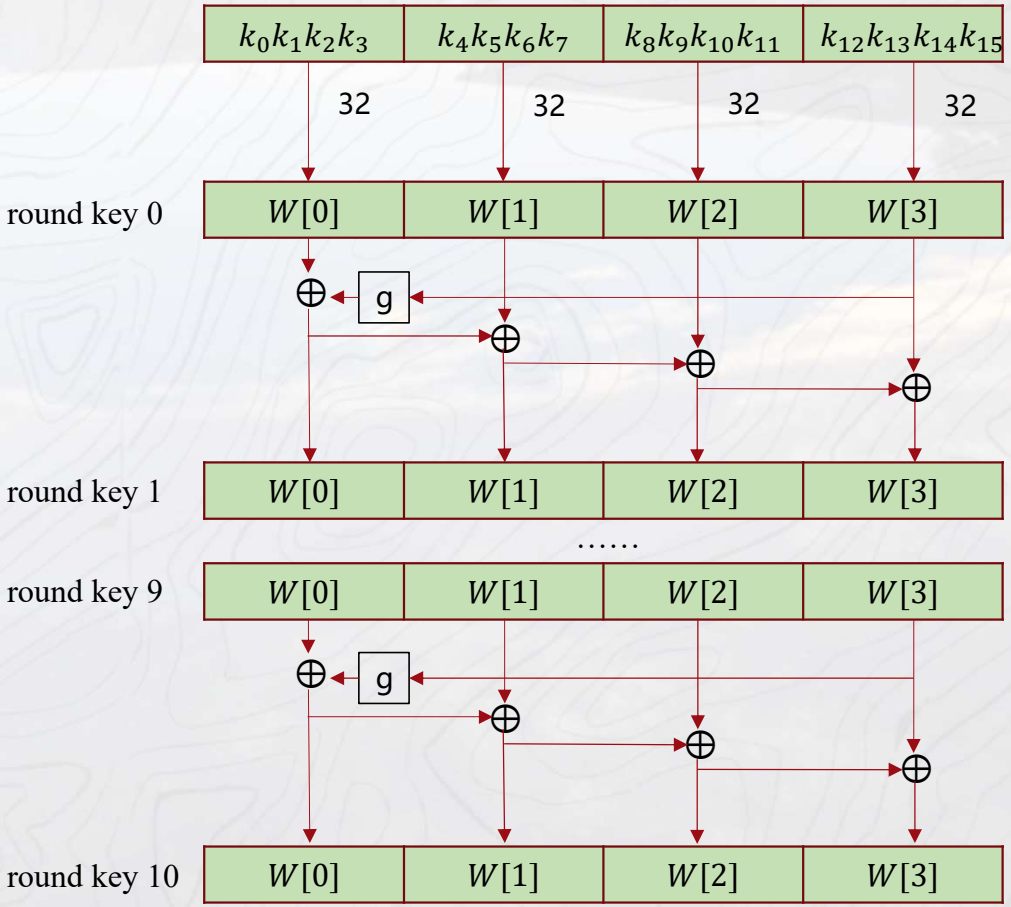
计算方式：
假设待加密矩阵的某个数为x
$x * 01$，为x本身；
$x * 02$，x的二进制左移一位（右边补0），如果溢出（即如果x的二进制最高位为1），那么再异或上1B；
$x * 03$，结果为 $(x * 02) \oplus x$，即，先乘02再异或本身，计算方法和上面一样。
例：02 * 87的计算过程
87的二进制位10000111，左移一位得到00001110，因为本身的二进制最高位为1，所以再异或1B即 $00001110 \oplus 00011011 = 00010101$，转换到16进制是15。

- **AES(Advanced Encryption Standard 高级加密标准)——轮函数**

3. 轮密钥异或

- 输入：16位矩阵C，16位密钥k_i
- 输出：$C \oplus k_i$
- 密钥生成方案：
  - 密钥扩展：将原来的4字、6字或者8字密钥扩展成拥有一定字数的长密钥。
  - 密钥选取：从长密钥中选取若干部分，使其充当AES每一轮迭代的轮密钥。

Key length (bits)	Number of subkeys
128	11
192	13
256	15



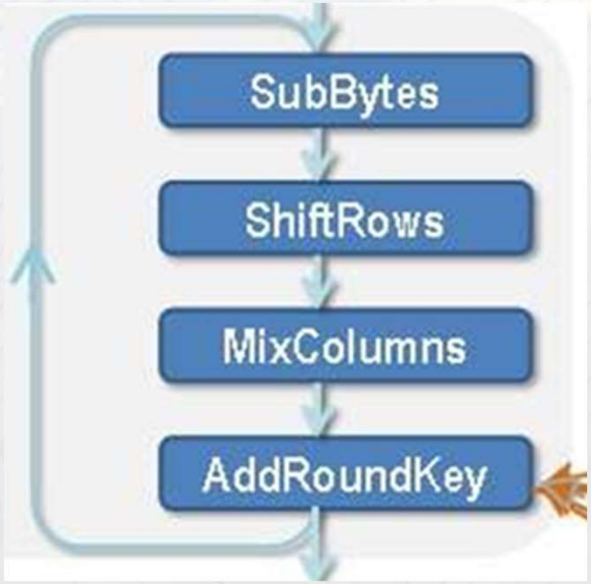
Word-oriented: 1 word = 4 byte = 32 bits

- **AES(Advanced Encryption Standard 高级加密标准)——轮函数**

例：待加密矩阵是 $\begin{bmatrix} EA & 04 & 65 & 85 \\ 83 & 45 & 5D & 96 \\ 5C & 33 & 02 & B0 \\ F0 & 2D & AD & C5 \end{bmatrix}$，列混淆矩阵是 $\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix}$，轮密钥是 $\begin{bmatrix} AC & 19 & 28 & 57 \\ 77 & FA & D1 & 5C \\ 66 & DC & 29 & 00 \\ F3 & 21 & 41 & 6A \end{bmatrix}$

S盒是

x \ y	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	63	7C	77	7B	F2	6B	6F	C5	30	01	67	2B	FE	D7	AB	76
1	CA	82	C9	7D	FA	59	47	F0	AD	D4	A2	AF	9C	A4	72	C0
2	B7	FD	93	26	36	3F	F7	CC	34	A5	E5	F1	71	D8	31	15
3	04	C7	23	C3	18	96	05	9A	07	12	80	E2	EB	27	B2	75
4	09	83	2C	1A	1B	6E	5A	A0	52	3B	D6	B3	29	E3	2F	84
5	53	D1	00	ED	20	FC	B1	5B	6A	CB	BE	39	4A	4C	58	CF
6	D0	EF	AA	FB	43	4D	33	85	45	F9	02	7F	50	3C	9F	A8
7	51	A3	40	8F	92	9D	38	F5	BC	B6	DA	21	10	FF	F3	D2
8	CD	0C	13	EC	5F	97	44	17	C4	A7	7E	3D	64	5D	19	73
9	60	81	4F	DC	22	2A	90	88	46	EE	B8	14	DE	5E	0B	DB
A	E0	32	3A	0A	49	06	24	5C	C2	D3	AC	62	91	95	E4	79
B	E7	C8	37	6D	8D	D5	4E	A9	6C	56	F4	EA	65	7A	AE	08
C	BA	78	25	2E	1C	A6	B4	C6	E8	DD	74	1F	4B	BD	8B	8A
D	70	3E	B5	66	48	03	F6	0E	61	35	57	B9	86	C1	1D	9E
E	E1	F8	98	11	69	D9	8E	94	9B	1E	87	E9	CE	55	28	DF
F	8C	A1	89	0D	BF	E6	42	68	41	99	2D	0F	B0	54	BB	16



- **AES(Advanced Encryption Standard 高级加密标准)——轮函数**

例：待加密矩阵是 $\begin{bmatrix} EA & 04 & 65 & 85 \\ 83 & 45 & 5D & 96 \\ 5C & 33 & 02 & B0 \\ F0 & 2D & AD & C5 \end{bmatrix}$，S盒是

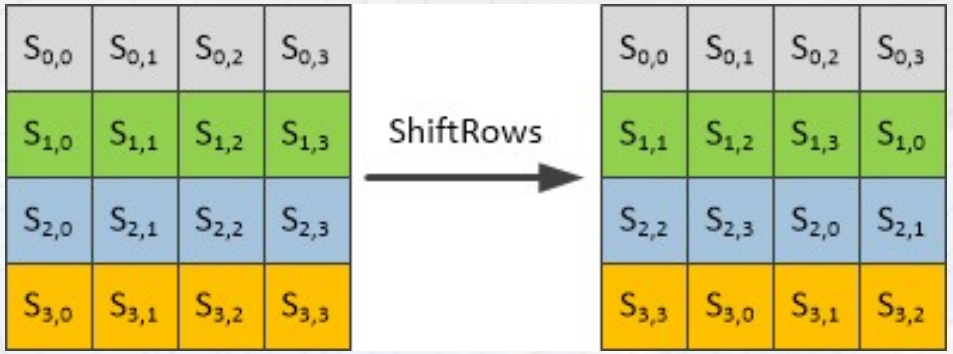
x \ y	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	63	7C	77	7B	F2	6B	6F	C5	30	01	67	2B	FE	D7	AB	76
1	CA	82	C9	7D	FA	59	47	F0	AD	D4	A2	AF	9C	A4	72	C0
2	B7	FD	93	26	36	3F	F7	CC	34	A5	E5	F1	71	D8	31	15
3	04	C7	23	C3	18	96	05	9A	07	12	80	E2	EB	27	B2	75
4	09	83	2C	1A	1B	6E	5A	A0	52	3B	D6	B3	29	E3	2F	84
5	53	D1	00	ED	20	FC	B1	5B	6A	CB	BE	39	4A	4C	58	CF
6	D0	EF	AA	FB	43	4D	33	85	45	F9	02	7F	50	3C	9F	A8
7	51	A3	40	8F	92	9D	38	F5	BC	B6	DA	21	10	FF	F3	D2
8	CD	0C	13	EC	5F	97	44	17	C4	A7	7E	3D	64	5D	19	73
9	60	81	4F	DC	22	2A	90	88	46	EE	B8	14	DE	5E	0B	DB
A	E0	32	3A	0A	49	06	24	5C	C2	D3	AC	62	91	95	E4	79
B	E7	C8	37	6D	8D	D5	4E	A9	6C	56	F4	EA	65	7A	AE	08
C	BA	78	25	2E	1C	A6	B4	C6	E8	DD	74	1F	4B	BD	8B	8A
D	70	3E	B5	66	48	03	F6	0E	61	35	57	B9	86	C1	1D	9E
E	E1	F8	98	11	69	D9	8E	94	9B	1E	87	E9	CE	55	28	DF
F	8C	A1	89	0D	BF	E6	42	68	41	99	2D	0F	B0	54	BB	16

1. 字节代换：

$$\begin{bmatrix} 87 & F2 & 4D & 97 \\ EC & 6E & 4C & 90 \\ 4A & C3 & 46 & E7 \\ 8C & D8 & 95 & A6 \end{bmatrix}$$

- **AES(Advanced Encryption Standard 高级加密标准)——轮函数**

例：字节代换后的矩阵是 $\begin{bmatrix} 87 & F2 & 4D & 97 \\ EC & 6E & 4C & 90 \\ 4A & C3 & 46 & E7 \\ 8C & D8 & 95 & A6 \end{bmatrix}$



2. 行位移：

$$\begin{bmatrix} 87 & F2 & 4D & 97 \\ 6E & 4C & 90 & EC \\ 46 & E7 & 4A & C3 \\ A6 & 8C & D8 & 95 \end{bmatrix}$$

- **AES(Advanced Encryption Standard 高级加密标准)——轮函数**

例：行位移后的矩阵是$\begin{bmatrix}87 & F2 & 4D & 97\\ 6E & 4C & 90 & EC\\ 46 & E7 & 4A & C3\\ A6 & 8C & D8 & 95\end{bmatrix}$，列混淆矩阵是$\begin{bmatrix}02 & 03 & 01 & 01\\ 01 & 02 & 03 & 01\\ 01 & 01 & 02 & 03\\ 03 & 01 & 01 & 02\end{bmatrix}$

3. 列混淆：

$$\begin{bmatrix}02 & 03 & 01 & 01\\ 01 & 02 & 03 & 01\\ 01 & 01 & 02 & 03\\ 03 & 01 & 01 & 02\end{bmatrix} * \begin{bmatrix}87 & F2 & 4D & 97\\ 6E & 4C & 90 & EC\\ 46 & E7 & 4A & C3\\ A6 & 8C & D8 & 95\end{bmatrix} = \begin{bmatrix}47 & 40 & A3 & 4C\\ 37 & D4 & 70 & 9F\\ 94 & E4 & 3A & 42\\ ED & A5 & A6 & BC\end{bmatrix}$$

以第一行第一列的计算为例：

$$02 * 87 \oplus 03 * 6\mathrm{E} \oplus 01 * 46 \oplus 01 * A6 = 15 \oplus B2 \oplus 46 \oplus A6 = 47$$

87（10000111）左移一位得到0000111，最高位是1，再与1B异或，得到15
6E（01101110）先左移得到11011100 再与本身异或，得到B2
2个46不变

- **AES(Advanced Encryption Standard 高级加密标准)——轮函数**

例：列混淆后的矩阵是 $\begin{bmatrix} 47 & 40 & A3 & 4C \\ 37 & D4 & 70 & 9F \\ 94 & E4 & 3A & 42 \\ ED & A5 & A6 & BC \end{bmatrix}$，轮密钥是 $\begin{bmatrix} AC & 19 & 28 & 57 \\ 77 & FA & D1 & 5C \\ 66 & DC & 29 & 00 \\ F3 & 21 & 41 & 6A \end{bmatrix}$

4. 轮密钥异或：

$$\begin{bmatrix} AC & 19 & 28 & 57 \\ 77 & FA & D1 & 5C \\ 66 & DC & 29 & 00 \\ F3 & 21 & 41 & 6A \end{bmatrix} \oplus \begin{bmatrix} 47 & 40 & A3 & 4C \\ 37 & D4 & 70 & 9F \\ 94 & E4 & 3A & 42 \\ ED & A5 & A6 & BC \end{bmatrix} = \begin{bmatrix} EB & 59 & 8B & 1B \\ 40 & 2E & A1 & C3 \\ F2 & 38 & 13 & 42 \\ 1E & 84 & E7 & D6 \end{bmatrix}$$

AC	10101100
47	01000111
XOR	11101011→EB

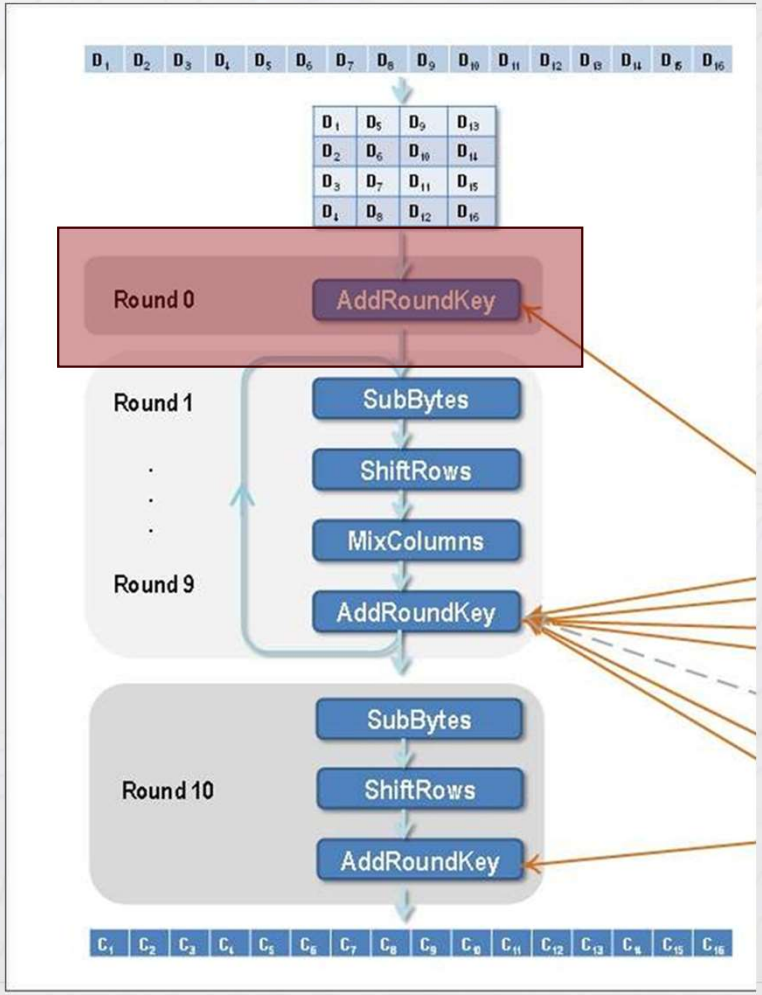
- **AES(Advanced Encryption Standard 高级加密标准)——练习**

练习：待加密矩阵 $\begin{bmatrix} 32 & 88 & 31 & E0 \\ 43 & 5A & 31 & 37 \\ F6 & 30 & 98 & 07 \\ A8 & 8D & A2 & 34 \end{bmatrix}$

轮密钥 $\begin{bmatrix} 2B & 28 & AB & 09 \\ 7E & AE & F7 & CF \\ 15 & D2 & 15 & 4F \\ 16 & A6 & 88 & 3C \end{bmatrix}$

明文	32	0011 0010
密钥	2B	0010 1011
结果	19	0001 1001

轮密钥异或的结果 $\begin{bmatrix} 19 & A0 & 9A & E9 \\ 3D & F4 & C6 & F8 \\ E3 & E2 & 8D & 48 \\ BE & 2B & 2A & 08 \end{bmatrix}$

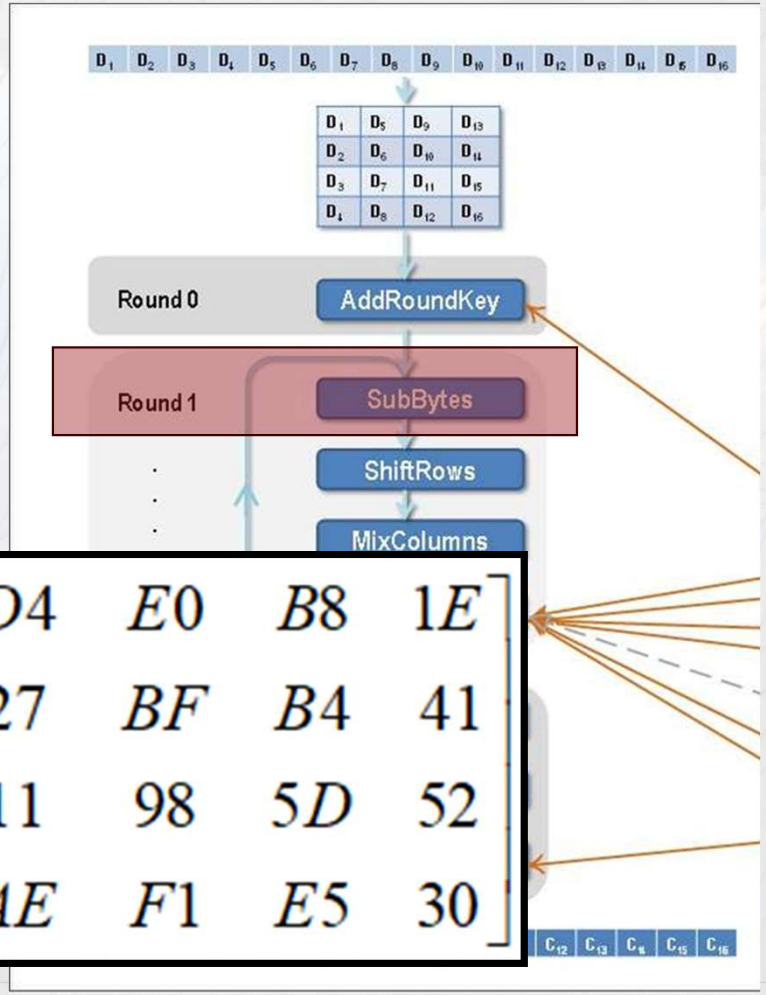


- **AES(Advanced Encryption Standard 高级加密标准)——练习**

练习：轮密钥异或的结果

$$\begin{bmatrix} 19 & A0 & 9A & E9 \\ 3D & F4 & C6 & F8 \\ E3 & E2 & 8D & 48 \\ BE & 2B & 2A & 08 \end{bmatrix}$$

x \ y	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	63	7C	77	7B	F2	6B	6F	C5	30	01	67	2B	FE	D7	AB	76
1	CA	82	C9	7D	FA	59	47	F0	AD	D4	A2	AF	9C	A4	72	C0
2	B7	FD	93	26	36	3F	F7	CC	34	A5	E5	F1	71	D8	31	15
3	04	C7	23	C3	18	96	05	9A	07	12	80	E2	EB	27	B2	75
4	09	83	2C	1A	1B	6E	5A	A0	52	3B	D6	B3	29	E3	2F	84
5	53	D1	00	ED	20	FC	B1	5B	6A	CB	BE	39	4A	4C	58	CF
6	D0	EF	AA	FB	43	4D	33	85	45	F9	02	7F	50	3C	9F	A8
7	51	A3	40	8F	92	9D	38	F5	BC	B6	DA	21	10	FF	F3	D2
8	CD	0C	13	EC	5F	97	44	17	C4	A7	7E	3D	64	5D	19	73
9	60	81	4F	DC	22	2A	90	88	46	EE	B8	14	DE	5E	0B	DB
A	E0	32	3A	0A	49	06	24	5C	C2	D3	AC	62	91	95	E4	79
B	E7	C8	37	6D	8D	D5	4E	A9	6C	56	F4	EA	65	7A	AE	08
C	BA	78	25	2E	1C	A6	B4	C6	E8	DD	74	1F	4B	BD	8B	8A
D	70	3E	B5	66	48	03	F6	0E	61	35	57	B9	86	C1	1D	9E
E	E1	F8	98	11	69	D9	8E	94	9B	1E	87	E9	CE	55	28	DF
F	8C	A1	89	0D	BF	E6	42	68	41	99	2D	0F	B0	54	BB	16

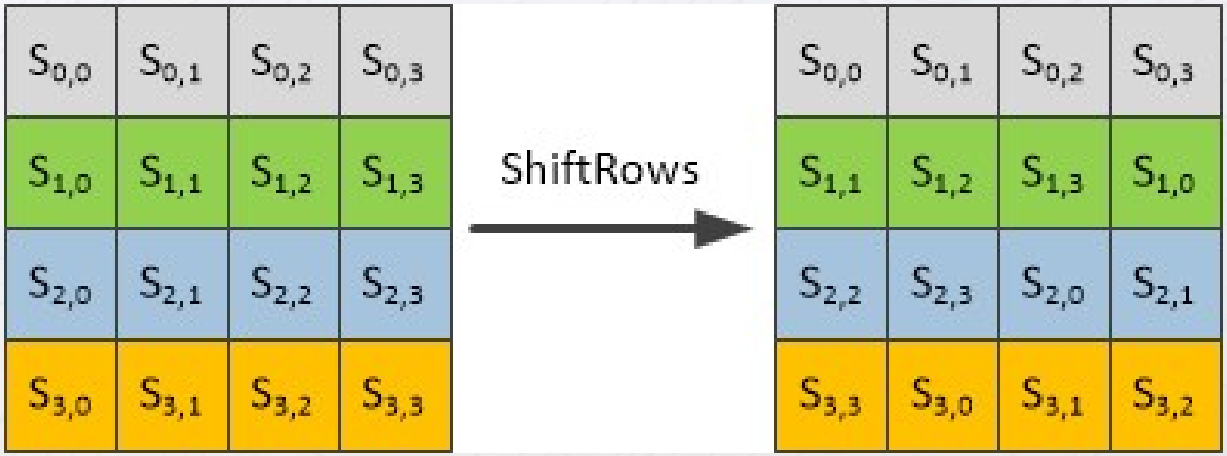


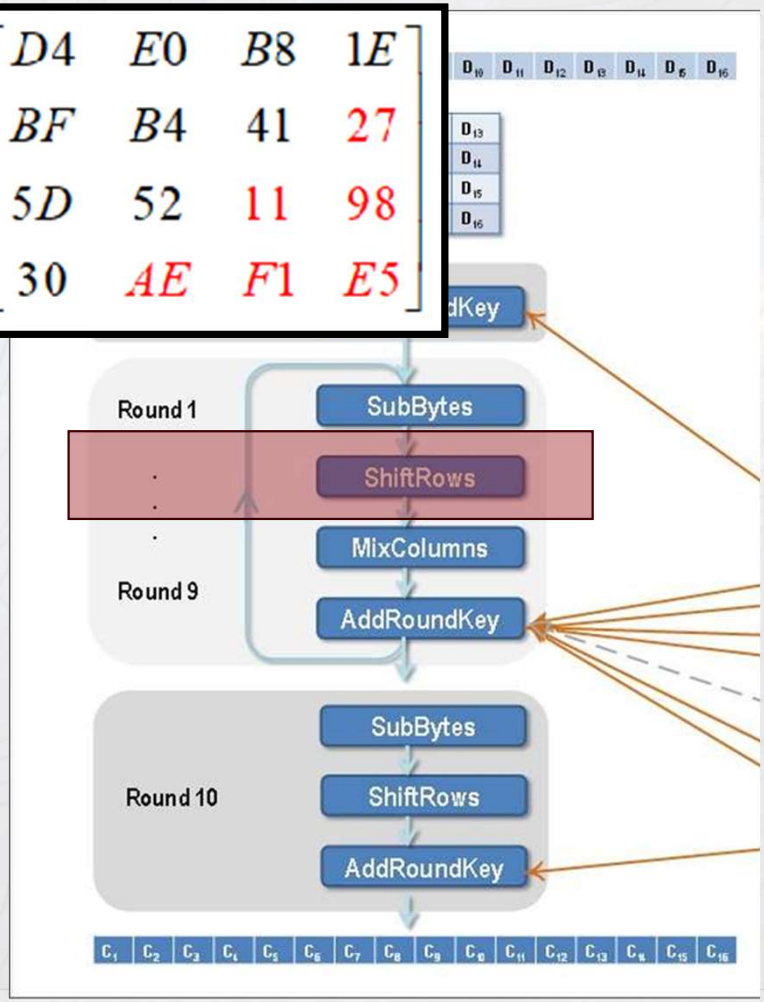
$$\begin{bmatrix} D4 & E0 & B8 & 1E \\ 27 & BF & B4 & 41 \\ 11 & 98 & 5D & 52 \\ AE & F1 & E5 & 30 \end{bmatrix}$$

- **AES(Advanced Encryption Standard 高级加密标准)——练习**

练习：字节代换后

$$\begin{bmatrix} D4 & E0 & B8 & 1E \\ 27 & BF & B4 & 41 \\ 11 & 98 & 5D & 52 \\ AE & F1 & E5 & 30 \end{bmatrix} \rightarrow \begin{bmatrix} D4 & E0 & B8 & 1E \\ BF & B4 & 41 & 27 \\ 5D & 52 & 11 & 98 \\ 30 & AE & F1 & E5 \end{bmatrix}$$





- **AES(Advanced Encryption Standard 高级加密标准)——练习**

练习：行位移后

$$\begin{bmatrix} D4 & E0 & B8 & 1E \\ BF & B4 & 41 & 27 \\ 5D & 52 & 11 & 98 \\ 30 & AE & F1 & E5 \end{bmatrix}$$

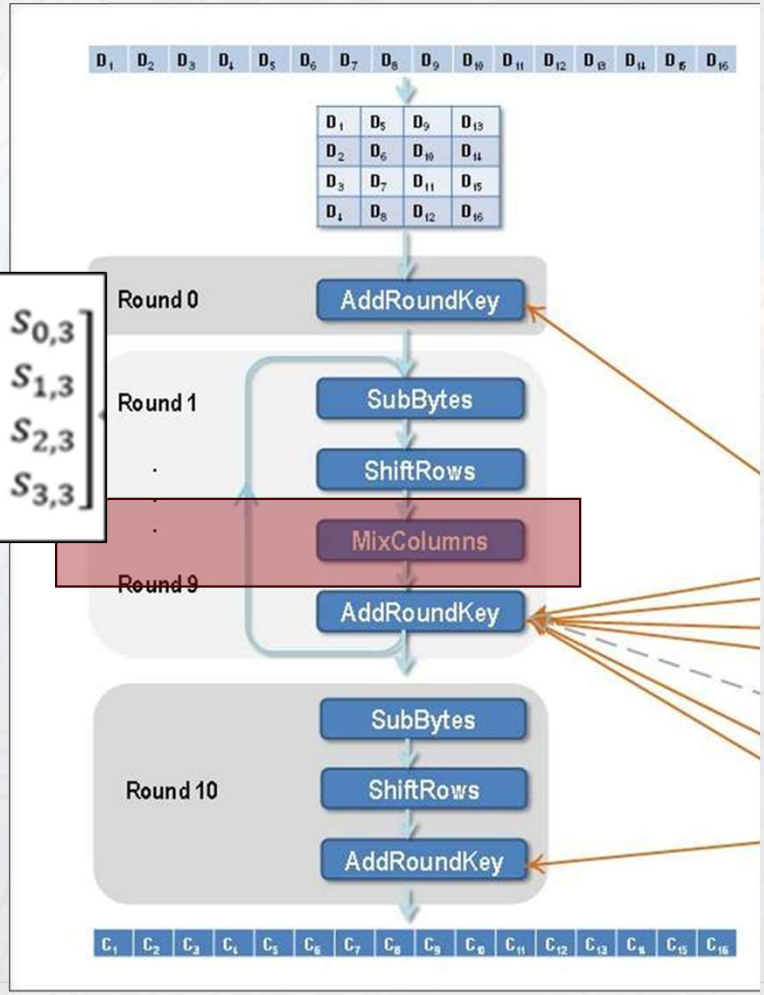
$$\begin{bmatrix} S'_{0,0} & S'_{0,1} & S'_{0,2} & S'_{0,3} \\ S'_{1,0} & S'_{1,1} & S'_{1,2} & S'_{1,3} \\ S'_{2,0} & S'_{2,1} & S'_{2,2} & S'_{2,3} \\ S'_{3,0} & S'_{3,1} & S'_{3,2} & S'_{3,3} \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} S_{0,0} & S_{0,1} & S_{0,2} & S_{0,3} \\ S_{1,0} & S_{1,1} & S_{1,2} & S_{1,3} \\ S_{2,0} & S_{2,1} & S_{2,2} & S_{2,3} \\ S_{3,0} & S_{3,1} & S_{3,2} & S_{3,3} \end{bmatrix}$$

$$S'_{0,j} = S_{0,j} \times 02 \oplus S_{1,j} \times 03 \oplus S_{2,j} \times 01 \oplus S_{3,j} \times 01$$

$$S'_{1,j} = S_{0,j} \times 01 \oplus S_{1,j} \times 02 \oplus S_{2,j} \times 03 \oplus S_{3,j} \times 01$$

$$S'_{2,j} = S_{0,j} \times 01 \oplus S_{1,j} \times 01 \oplus S_{2,j} \times 02 \oplus S_{3,j} \times 03$$

$$S'_{3,j} = S_{0,j} \times 03 \oplus S_{1,j} \times 01 \oplus S_{2,j} \times 01 \oplus S_{3,j} \times 02$$



- **AES(Advanced Encryption Standard 高级加密标准)——练习**

练习：行位移后 $\begin{bmatrix} D4 & E0 & B8 & 1E \\ BF & B4 & 41 & 27 \\ 5D & 52 & 11 & 98 \\ 30 & AE & F1 & E5 \end{bmatrix}$

$$\begin{bmatrix} S'_{0,0} & S'_{0,1} & S'_{0,2} & S'_{0,3} \\ S'_{1,0} & S'_{1,1} & S'_{1,2} & S'_{1,3} \\ S'_{2,0} & S'_{2,1} & S'_{2,2} & S'_{2,3} \\ S'_{3,0} & S'_{3,1} & S'_{3,2} & S'_{3,3} \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} S_{0,0} & S_{0,1} & S_{0,2} & S_{0,3} \\ S_{1,0} & S_{1,1} & S_{1,2} & S_{1,3} \\ S_{2,0} & S_{2,1} & S_{2,2} & S_{2,3} \\ S_{3,0} & S_{3,1} & S_{3,2} & S_{3,3} \end{bmatrix}$$

$$S'_{0,0} = 2 \times D4 \oplus 3 \times BF \oplus 1 \times 5D \oplus 1 \times 30$$

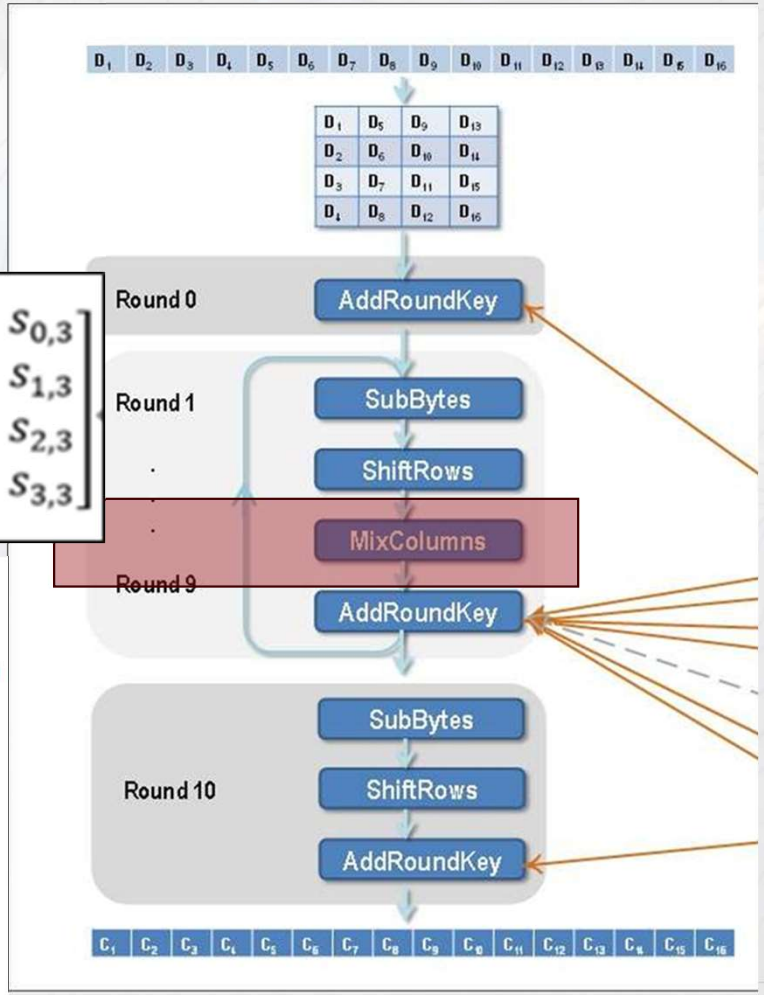
$$S'_{0,0} = 2 \times (11010100) \oplus 3 \times (10111111) \oplus (01011101) \oplus (00110000)$$

$$S'_{0,0} = (10101000)_{[D4左移]} \oplus (00011011)_{[1B]}$$

$$\oplus (01111110)_{[BF左移]} \oplus (00011011)_{[1B]} \oplus (10111111)_{[BF]}$$

$$\oplus (01011101)_{[5D]} \oplus (00110000)_{[30]}$$

$$S'_{0,0} = 00000100 = 04$$



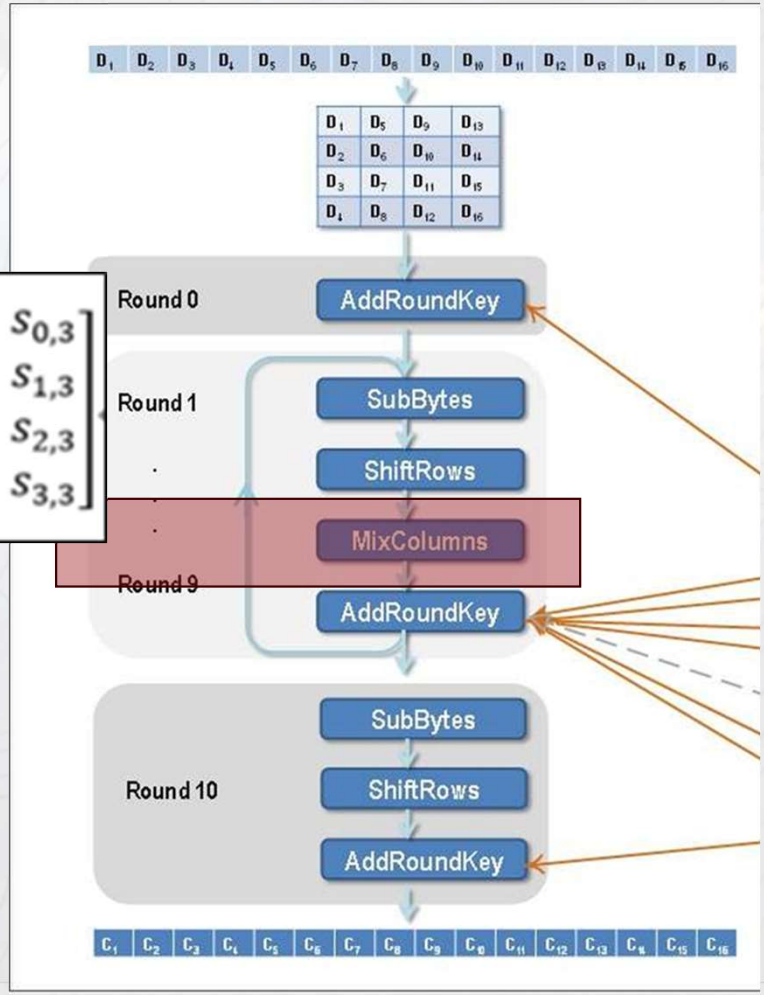
- **AES(Advanced Encryption Standard 高级加密标准)——练习**

练习：行位移后

$$\begin{bmatrix} D4 & E0 & B8 & 1E \\ BF & B4 & 41 & 27 \\ 5D & 52 & 11 & 98 \\ 30 & AE & F1 & E5 \end{bmatrix}$$

$$\begin{bmatrix} S'_{0,0} & S'_{0,1} & S'_{0,2} & S'_{0,3} \\ S'_{1,0} & S'_{1,1} & S'_{1,2} & S'_{1,3} \\ S'_{2,0} & S'_{2,1} & S'_{2,2} & S'_{2,3} \\ S'_{3,0} & S'_{3,1} & S'_{3,2} & S'_{3,3} \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} S_{0,0} & S_{0,1} & S_{0,2} & S_{0,3} \\ S_{1,0} & S_{1,1} & S_{1,2} & S_{1,3} \\ S_{2,0} & S_{2,1} & S_{2,2} & S_{2,3} \\ S_{3,0} & S_{3,1} & S_{3,2} & S_{3,3} \end{bmatrix}$$

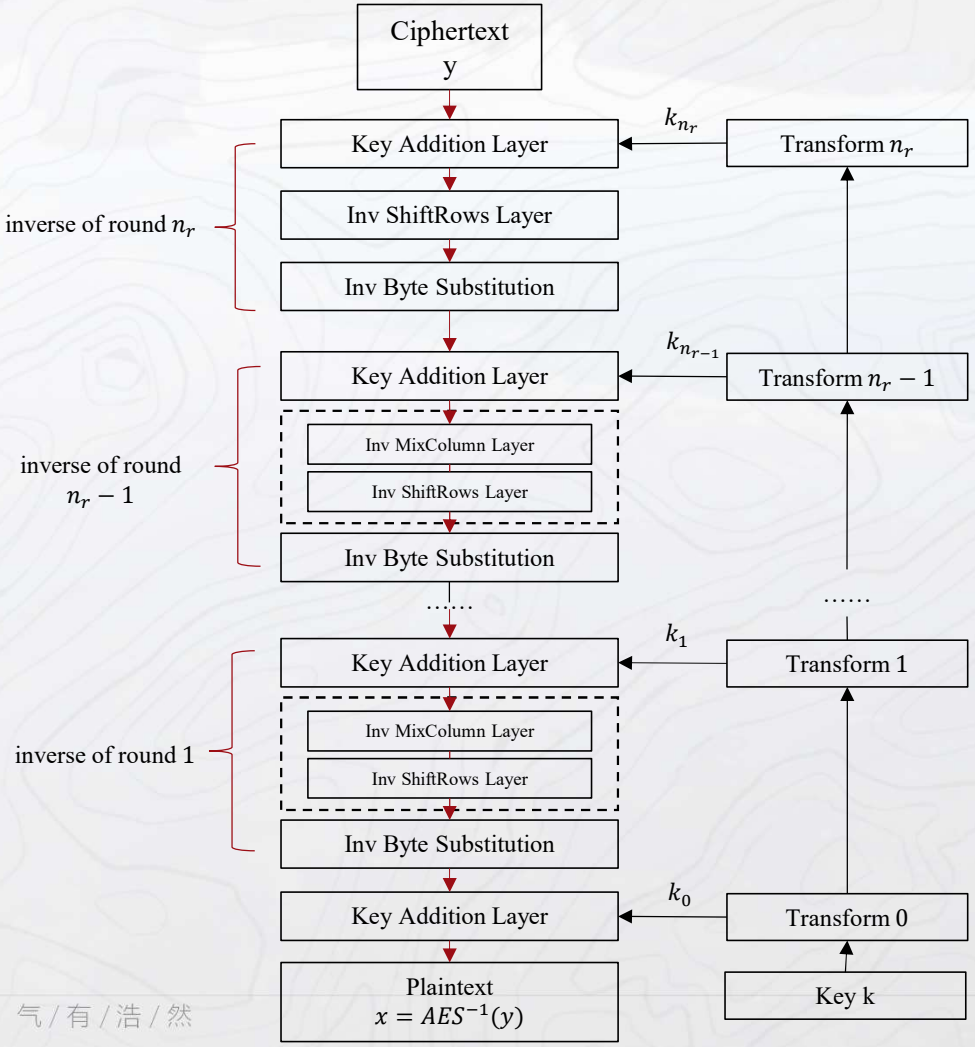
$$\begin{bmatrix} 04 & E0 & 48 & 28 \\ 66 & CB & F8 & 06 \\ 81 & 19 & D3 & 26 \\ E5 & 9A & 7A & 4C \end{bmatrix}$$



- **AES(Advanced Encryption Standard 高级加密标准)**

	Code size	Performance
Pre-compute round functions (24KB or 4KB)	largest	fastest: table lookups and xors
Pre-compute S-box only (256 bytes)	smaller	slower
No pre-computation	smallest	slowest

- **AES(Advanced Encryption Standard 高级加密标准)——解密**



- AES is not based on a Feistel network
  ⇒ All layers must be inverted for decryption:
- MixColumn layer → **Inv MixColumn layer**
- ShiftRows layer→ **Inv ShiftRows layer**
- Byte Substitution layer → **Inv Byte Substitution layer**
- Key Addition layer is its own inverse

- **AES(Advanced Encryption Standard 高级加密标准)——解密**

逆向列混淆:

- To reverse the MixColumn operation, each column of the state matrix C must be multiplied with the **inverse of the 4x4 matrix**, e.g.,

$$\begin{pmatrix} B_0 \\ B_1 \\ B_2 \\ B_3 \end{pmatrix} = \begin{pmatrix} 0E & 0B & 0D & 09 \\ 09 & 0E & 0B & 0D \\ 0D & 09 & 0E & 0B \\ 0B & 0D & 09 & 0E \end{pmatrix} \cdot \begin{pmatrix} C_0 \\ C_1 \\ C_2 \\ C_3 \end{pmatrix}$$

where 09, $0B$, $0D$ and $0E$ are given in hexadecimal notation

- Again, all arithmetic is done in the Galois field $GF(256)$

- **AES(Advanced Encryption Standard 高级加密标准)——解密**

逆向行位移:

Input matrix

B_0	B_4	B_8	B_{12}
B_1	B_5	B_9	B_{13}
B_2	B_6	B_{10}	B_{14}
B_3	B_7	B_{11}	B_{15}

Output matrix

B_0	B_4	B_8	B_{12}	no shift
B_{13}	B_1	B_5	B_9	← one position right shift
B_{10}	B_{14}	B_2	B_6	← two position right shift
B_7	B_{11}	B_{15}	B_3	← three position right shift

- **AES(Advanced Encryption Standard 高级加密标准)——解密**

逆向S盒:

52	9	6a	d5	30	36	a5	38	bf	40	a3	9e	81	f3	d7	fb
7c	e3	39	82	9b	2f	ff	87	34	8e	43	44	c4	de	e9	cb
54	7b	94	32	a6	c2	23	3d	ee	4c	95	0b	42	fa	c3	4e
8	2e	a1	66	28	d9	24	b2	76	5b	a2	49	6d	8b	d1	25
72	f8	f6	64	86	68	98	16	d4	a4	5c	cc	5d	65	b6	92
6c	70	48	50	fd	ed	b9	da	5e	15	46	57	a7	8d	9d	84
90	d8	ab	0	8c	bc	d3	0a	f7	e4	58	5	b8	b3	45	6
d0	2c	1e	8f	ca	3f	0f	2	c1	af	bd	3	1	13	8a	6b
3a	91	11	41	4f	67	dc	ea	97	f2	cf	ce	f0	b4	e6	73
96	ac	74	22	e7	ad	35	85	e2	f9	37	e8	1c	75	df	6e
47	f1	1a	71	1d	29	c5	89	6f	b7	62	0e	aa	18	be	1b
fc	56	3e	4b	c6	d2	79	20	9a	db	c0	fe	78	cd	5a	f4
1f	dd	a8	33	88	7	c7	31	b1	12	10	59	27	80	ec	5f
60	51	7f	a9	19	b5	4a	0d	2d	e5	7a	9f	93	c9	9c	ef
a0	e0	3b	4d	ae	2a	f5	b0	c8	eb	bb	3c	83	53	99	61
17	2b	4	7e	ba	77	d6	26	e1	69	14	63	55	21	0c	7d

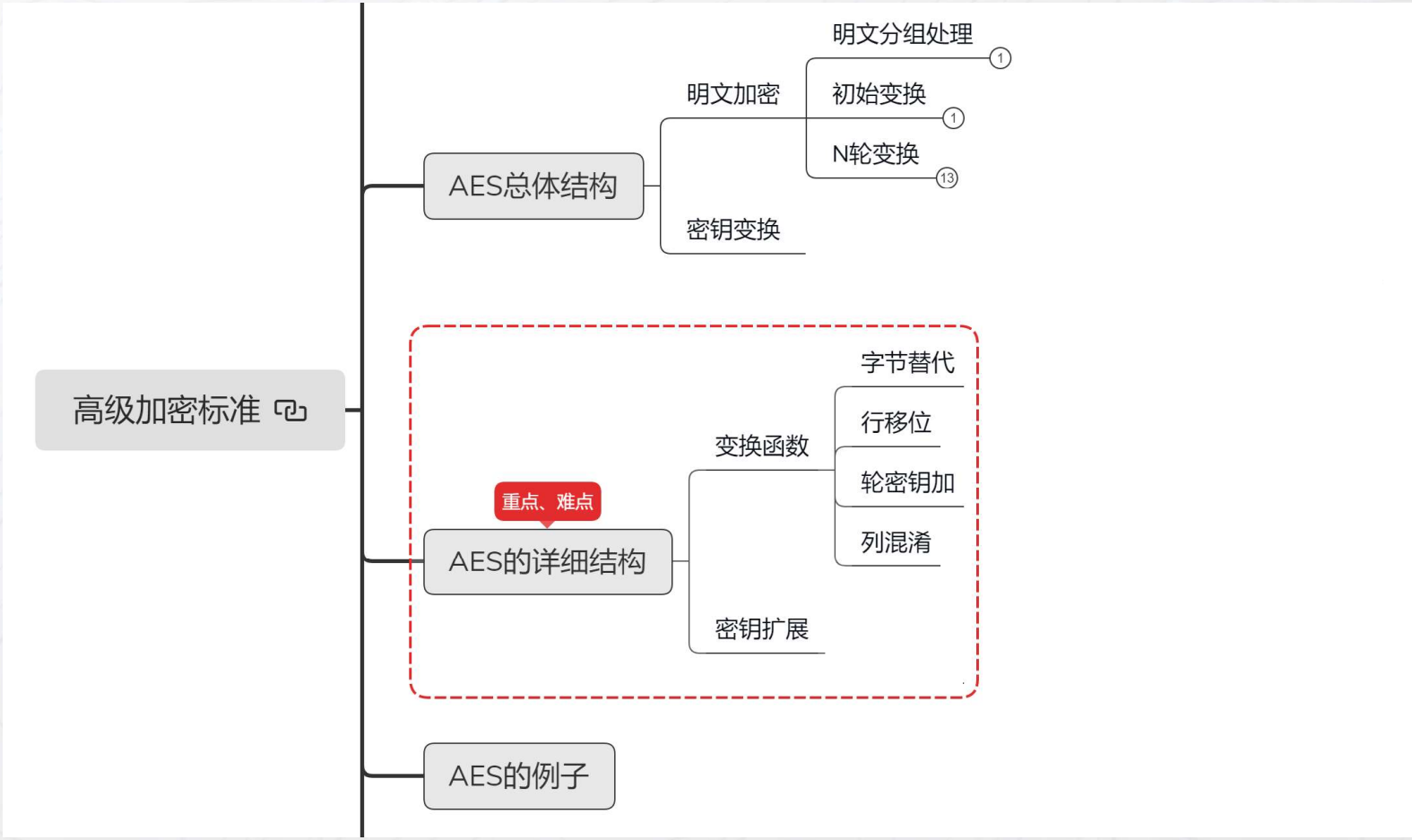
逆向轮密钥异或:
密钥与加密一致

- **AES(Advanced Encryption Standard 高级加密标准)**

总结

- AES 是一种现代分组密码，支持 128、192 和 256 位三种密钥长度。它能提供出色的长期安全性，抵御暴力破解攻击。
- 自 20 世纪 90 年代末以来，人们对 AES 进行了深入研究，但尚未发现比暴力破解更好的攻击方式。
- AES 并非基于费斯特尔网络。它的基本操作使用伽罗瓦域，具有很强的扩散性和混淆性。
- AES 是许多开放标准（如 IPsec 或 TLS）的一部分，此外还是美国政府应用的强制加密算法。在未来的许多年里，该加密算法都有可能成为主流加密算法。
- AES 在软件和硬件方面都很高效 。

- **AES(Advanced Encryption Standard 高级加密标准)**



- **AES(Advanced Encryption Standard 高级加密标准)——解密**

Pentium 4, 2.1 GHz (on Windows XP SP1, Visual C++ 2003)

	Cipher	Block/key size	Speed (MB/sec)
Stream cipher	RC4		113
	SEAL		293
Block cipher	3DES	64/168	9
	AES	128/128	61
	IDEA	64/128	19
	SHACAL-2	512/128	20


Any Questions?