



计算机网络安全技术

- 课程代号：40240572
- 课程对象：本科生
- 授课教师：尹 霞
- 开课单位：计算机系网络所

密码学基础

初识密码

- ①与人类共生
- ②密码基本概念
- ③密码发展历程

古典密码

- ① 代换技术
- ② 置换技术
- ③ 破译举例

对称密码算法

- ① S-DES算法
- ② Feistel密码结构
- ③ DES算法
- ④ 常用对称密码

非对称密码算法

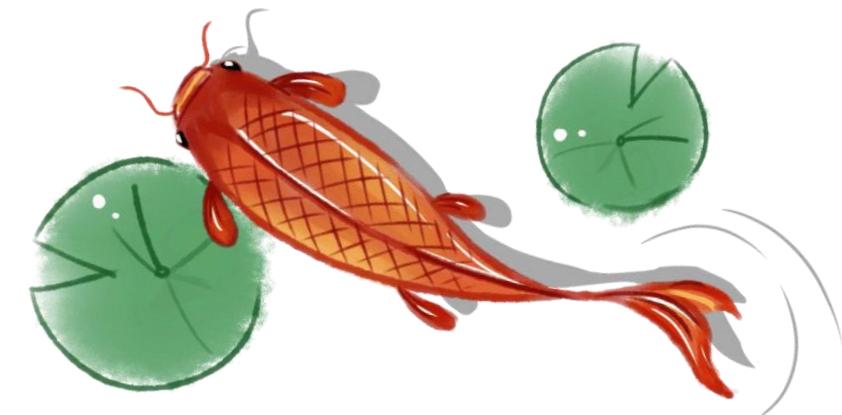
- ① 公钥密码原理
- ② 数论基础
- ③ RSA算法
- ④ DH密钥交换算法

密钥分配

- ①概述
- ②三个不同情况



初识密码[1]: 与人类共生



初识密码

- 密码是一种用来混淆的技术，它希望将正常的、可识别的信息转变为无法识别的信息
 - 密码学（Cryptology）一字源自希腊文“krypto’s”及“logos”两字，直译即为“隐藏”“讯息”之意
- 密码的历史几乎跟文字的历史一样长，自从有了文字以来，人们为了某种需要总是想法设法隐藏某些信息，以起到保证信息安全的目的
- 古时候的密码主要是用于军事和外交目的；随着商业的发展，在商贸等其它场合也开始使用密码
- 密码不同于隐藏：保险柜和一封密信是不同的
- 密码不同于访问控制：口令是访问控制，没有加密，安全性往往较弱

古代军事：军事密码本

- 北宋的曾公亮发明了我国的第一本军事密码本
- 他将常用的40个军事口令逐一编号，并用一首40个字的五言诗作为解密的钥匙，在出征前密告先锋官
 - 如将军口令为“18”，而五言诗中编号18的字为“戏”字，于是将军先随便写一封含有“戏”的信，然后在“戏”字上加盖印章作记号送出
 - 先锋官得令后默念一遍诗歌，即可从“戏”中数出命令号码
- 唐朝的五言诗不计其数，送信的人又不明就里，敌方即使严刑逼供，信使也是“打死说不出”

古代商业：平遥票号

- 清朝山西平遥钱庄票号就有自己完整的加密方法：银票中的密字
- 方法：用汉字作符号
 - 代表月：
 - 谨防假票冒取，勿忘细视书章
 - 代表日：
 - 堪笑事情薄，天道最公平，昧心图自利，
 - 阴谋害他人，善恶终有报，到头必分明
 - 代表数字：
 - 壹~拾：生客多察看，斟酌而后行
 - 万千百十：国宝流通

某人持一张汇票到北平票号，
汇票后写有
“**书薄 生国察宝多流**”，
伙计看后，就将银子交付来人。

密码案例：暗语

- 诗情画意：

- 早妆未罢暗凝眉，
- 迎户愁看紫燕飞，
- 无力回天春已老，
- 双栖画栋不如归。

- 刀光剑影

- 芦花丛中一扁舟，
- 俊杰俄从此地游，
- 义士若能知此理，
- 反躬难逃可无忧。

- 骂人：

- 日落香炉，舍去凡心一点。
- 炉熄火尽，务把意马牢栓。

- 王先生：

- 来信收悉，你的盛情真是难以报答。我已在昨天抵达广州。秋雨连绵，每天需备伞一把方能上街，苦矣。大约本月中旬我才能返回，届时再见。

密码案例：跳舞小人

- 福尔摩斯探案集----跳舞小人



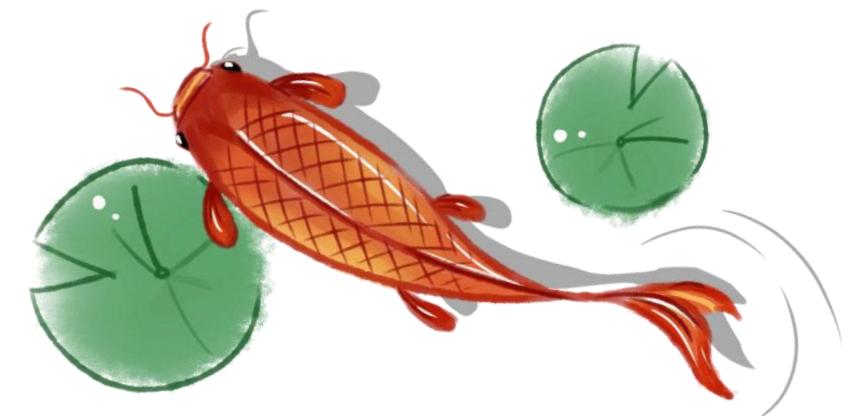
A		G		M		S		Y	
B		H		N		T		Z	
C		I		O		U			
D		J		P		V			
E		K		Q		W			
F		L		R		X			

隐写术

- 隐写术就是隐藏信息的存在，常见的方法有：
 - 字符标记、不可见墨水、针刺、打字机的色带校正
- 优点：
 - 能够被某些人使用不容易被发现
- 缺点：
 - 形式简单但构造费时，要求有大量的开销来隐藏相对较少的信息
 - 一旦构造方法被发现，就会变得完全没有价值
 - 隐写术一般没有稳健性



初始密码[2]: 密码学基本概念



两种加密形式

- 广泛使用的两种加密形式是传统加密和公钥加密
- 传统加密又称为对称加密、单钥加密
 - 代换密码、置换密码、二者组合
 - 缺陷：安全性在于保持算法本身的保密性
 - 不适合大规模生产、不适合较大的或者人员变动较大的组织
 - 用户无法了解算法的安全性
- 现代加密又称为非对称加密、公钥加密，于1976年第一次公开发表
 - 把算法和密钥分开，密码算法公开，密钥保密
 - 密码系统的安全性在于保持密钥的保密性，适于大规模生产

基本概念

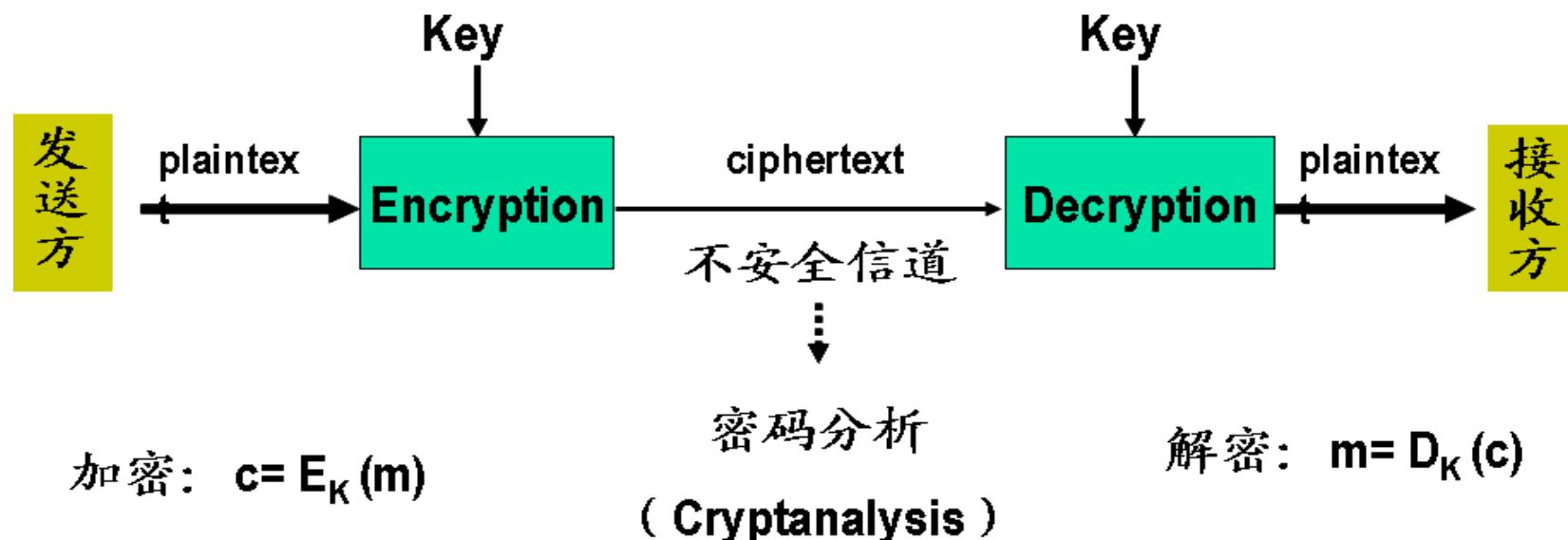
- 明文(plain text): 原始可以理解的信息或者数据
- 密文(cipher text): 加密后的信息
- 加密(encrypt, encryption): 从明文到密文的变换过程
- 解密(decrypt, decryption): 从密文到明文的变换过程

基本概念

- 密码算法 (Algorithm) /密码 (Cipher)：
 - 用来加密和解密的数学函数
 - $c=E(m)$, $m=D(c)$, $D(E(m))= m$
- 密钥(Key)
 - 密码算法中的一个变量
 - $c=E_{K_e}(m)$, $m=D_{K_d}(c)$, $D_{K_d}(E_{K_e}(m))= m$

密码学的基本模型

- 密码编码学(Cryptography): 研究各种加密方案的学科
- 密码分析学(Cryptanalysis): 研究破译密码获得消息的学科，又称“破译”
- 密码编码学和密码分析学统称为密码学



密码编码学

- 研究各种加密方案的学科称为密码编码学。
- 密码编码学系统具有三个独立的特征：
 - ① 转换明文为密文的运算类型
 - ② 所用的密钥数
 - ③ 处理明文的方法

密码编码学

- 密码编码学的特征一：转换明文为密文的运算类型

- 所有的加密算法基于两个原理：
置换(Transposition) 和 代换(Substitution)
- 置换是将明文中的元素重新排列
- 代换是将明文中每个元素映射成为另外一个元素
- 原则是不允许丢失信息，即所有的运算都是可逆的

密码编码学

- 密码编码学的特征二：所用的密钥数

- 发送方和接收方使用相同的密钥，称为对称密码、传统加密
- 发送方和接收方使用不相同的密钥，称为非对称密码、公钥密码

- 密码编码学的特征三：处理明文的方法

- 分组密码/块密码 (Block cipher):
每次处理一个输入分组，相应地输出一个输出分组
- 流密码/序列密码 (Stream cipher):
连续地处理输入元素，每次输出一个元素

一个简单的加密算法—异或

- 异或

$$0 \oplus 0 = 0$$

$$x \oplus 0 = x$$

$$1 \oplus 1 = 0$$

$$x \oplus x = 0$$

$$1 \oplus 0 = 1$$

$$x \oplus 1 = x^{-1}$$

$$0 \oplus 1 = 1$$

$$x \oplus x^{-1} = 1$$

异或运算（不带进位加法）：

明文:	\oplus	0	0	1	1
加密:		0	1	0	1
<hr/>					
密文:		0	1	1	0

$$C = P \oplus K$$

解密:

密文:	\oplus	0	1	1	0
密钥:		0	1	0	1
<hr/>					
明文:		0	0	1	1

$$P = C \oplus K$$

- 已知明文、密文，怎样求得密钥？

$$K = C \oplus P$$

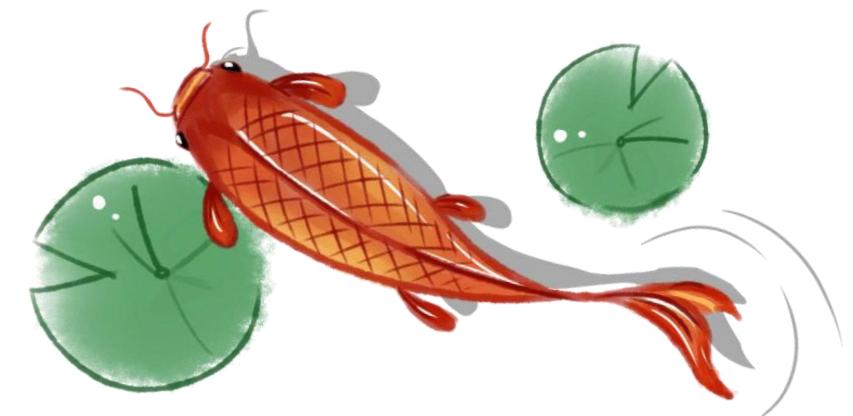
- 只知道密文，如何求得明文和密钥？

无条件安全和计算安全

- 无论有多少可以使用密文，都不足以唯一地确定由该体制产生密文所对应的明文，则加密机制是无条件安全的
- 一种加密机制在计算上是安全的，
当它满足以下条件之一： (computationally secure)
 - 破译密码的代价大于加密数据本身的价值
 - 破译密码的时间超过了密文信息的生命期



初始密码[3]: 密码学发展历程



密码学发展的三个阶段

- 古典密码（手工、机械阶段 ~1949年）

- 密码学是一门艺术

- 近代密码（计算机阶段：1949~1975）

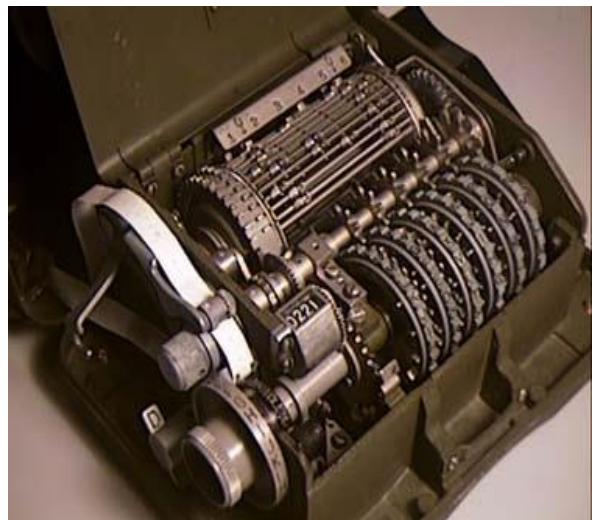
- 密码学成为科学

- 现代密码（1976~现在）

- 密码学的新方向 --- 公钥密码学

第1阶段：古典密码

- 古典密码（1949年前）：数据的安全基于算法的保密
 - 密码学还不是科学，而是艺术
- 在1949年之前，出现一些密码算法和加密设备；密码算法的基本手段是代换和置换，都是针对字符的；同时，还出现一些简单的密码分析手段



<http://hem.passagen.se/tan01/>



第2阶段：近代密码

- 计算机使得基于复杂计算的密码成为可能
- 有线电报产生了现代编码学
 - 1844年，萨米尔·莫尔斯发明了莫尔斯电码：用一系列的电子点划来进行电报通讯
 - 电报的出现第一次使远距离快速传递信息成为可能，增强了西方各国的通讯能力
- 无线电报产生了现代密码分析学
 - 20世纪初，意大利物理学家奎里亚摩·马可尼发明了无线电报，让无线电波成为新的通讯手段，它实现了远距离通讯的即时传输
 - 由于通过无线电波送出的每条信息不仅传给了己方，也传给了敌方，这就意味着必须给每条信息加密法的保密

第2阶段：近代密码

- 1883年，Kerchoffs给出了现代密码学的原理：
加密体系的安全性并不依赖于加密的方法本身，而是依赖于所使用的密匙
 - 这一原则已得到普遍承认，成为判定密码强度的衡量标准，实际上也成为古典密码和现代密码的分界线
- 1949年，香农(Claude Elwood Shannon)发表了“保密通信的信息理论”论文，使密码学成为一门科学
 - 当时的密码研究主要属于军事领域，密码人员都在黑屋子里工作，处于保密状态
 - 1949~1976年，近20年里密码的理论和技术没有大的进展

第3阶段：现代密码

- 1976年之后，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，该对称加密算法完全公开，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

第3阶段：现代密码

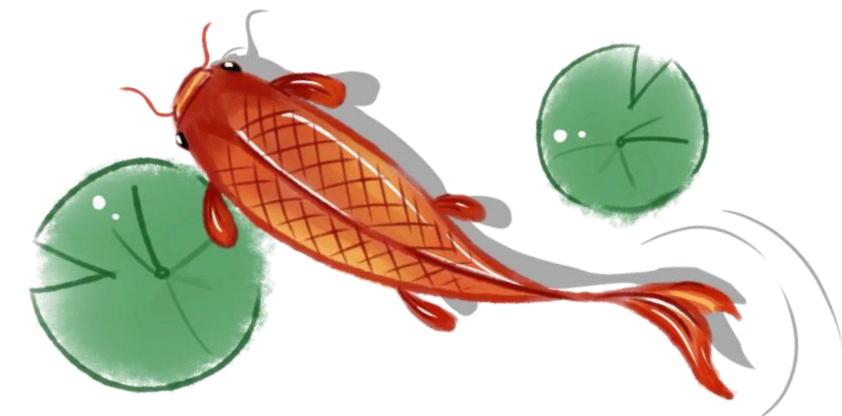
- 对称密码算法日趋成熟
 - 1977年，美国正式公布实施数据加密标准DES
 - 80年代，出现“过渡性”的后DES算法，如IDEA、RCx、CAST等
 - 90年代，对称密钥密码进一步成熟，Rijndael、RC6、MARS、Twofish、Serpent等出现
 - 1997年，美国国家标准技术研究所（NIST）发起了征集高级加密标准算法AES的活动
 - 2001年，Rijndael成为DES的替代者(AES)
 - 欧洲也耗巨资完成了“欧洲密码大计划”，制定自己的标准

第3阶段：现代密码

- 革命性的公钥密码算法出现，公钥密码使得发送端和接收端无密钥传输的保密通信成为可能
 - 1976年：Diffie & Hellman 的 “New Directions in Cryptography” 提出了不对称密钥算法（公钥密码算法）
 - 1977年，Rivest、Shamir、Adleman三人提出了RSA公钥算法
 - 90年代，逐步出现椭圆曲线等其他公钥算法



古典密码[1]: 代换技术



代换技术Substitution Cipher

- 代换技术是将明文字符替换成其他字母、数字或者符号的方法
 - Caesar密码
 - 单表代换密码
 - Playfair密码
 - Hill密码
 - Vigenere密码
 - Verman密码和一次一密

Caesar 密码

- 最早的代换密码是由Julius Caesar发明的；原理是对字母表中的每个字母用它之后的第3个字母代换
 - Caesar 密码 : $c = (m+3) \text{ Mod } 26$

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

- 可以采用穷举攻击破译Caesar 密码，因为
 - 已知Caesar 密码的加密和解密算法
 - 只需要测试25个密钥
 - 明文所用的语言是已知的，意义易于识别

单表代换→多表代换

- 密钥词密码：设一个密钥词放在前面，其余字母按顺序

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

- Caesar密码和密钥词密码都是单表代换密码
 - 密文是26个字母的任意置换，即从明文字母到密文字母的映射
 - 密文中还是带有原始字母使用频率的一些统计学特征，较易被攻破
- 代换密码必须考虑：明文的语法模式和结构有多少仍然保留在密文中，减少这种保留信息的方法有：
 - 对明文中的多个字母一起加密
 - 采用多表代换密码

Playfair密码

- Playfair密码是最著名的多表代换密码，它把明文中的双字母作为一个单元并将其转换成密文的双字母音节
- Playfair密码基于一个由密钥词构成的 5×5 字母矩阵
 - 举例：密钥词是monarchy
- 构造密钥词的方法是：
 - 密钥词从左至右、从上至下填在矩阵格里，再将剩余的字母按照字母表的顺序从左至右、从上至下填在剩余的格子里
 - 字母I和J当作一个字母

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

Playfair密码

- 对明文按照如下规则一次加密两个字母
- 如果该字母对是两个相同的字母，添加一个填充字
 - 例如x，先把balloon变成 ba lx lo on；4个字母对
- 落在矩阵同一行的明文字母对中的字母，由其右边的字母代换。
 - 例如：ar被代换为RM
- 落在矩阵同一列的明文字母对中字母，由其下面的字母来代换
 - 例如：mu被代换成CM
- 其它的明文字母对中字母按照如下方法代换：它所在的行是该字母所在的行，列是另外一个字母所在的列
 - 例如：hs被代换为BP

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

Playfair密码

- Playfair密码相对于简单的单表密码是巨大的进步
 - 因为 26×26 个字母对，比单字母判定要困难得多
- 在很长一段时间内，Playfaire密码被认为是牢不可破的，并且在一战和二战中被大量使用
- 但是Playfair密码依然完整地保留了明文语言的结构，几百个字母密文就足够分析出规律了

Hill密码

- Hill密码是1929年数学家Lester Hill发明的一个多表代换密码
- 密码算法是将m个连续的明文字符代换成m个密文字符。这是由m个线性等式决定的，在等式里每个字母被指定为一个数字（ $a=0, b=1, \dots, z=25$ ）。
 - 例如： $m=3$ ，系统可以描述为：
 - $c_1 = (k_{11}p_1 + k_{12}p_2 + k_{13}p_3) \bmod 26$
 - $c_2 = (k_{21}p_1 + k_{22}p_2 + k_{23}p_3) \bmod 26$
 - $c_3 = (k_{31}p_1 + k_{32}p_2 + k_{33}p_3) \bmod 26$

Hill密码

- 用矩阵表示 $C=KP \bmod 26$
 - C 是密文， K 是 3×3 的加密密钥矩阵， P 是明文

$$\begin{bmatrix} C_1 \\ C_2 \\ C_3 \end{bmatrix} = \begin{bmatrix} k_{11} & k_{12} & k_{13} \\ k_{11} & k_{12} & k_{13} \\ k_{11} & k_{12} & k_{13} \end{bmatrix} \begin{bmatrix} P_1 \\ P_2 \\ P_3 \end{bmatrix} \bmod 26$$

- 例：对明文“paymoremoney”用密钥K加密

$$K = \begin{pmatrix} 17 & 17 & 5 \\ 21 & 18 & 21 \\ 2 & 2 & 19 \end{pmatrix}$$

$$\begin{aligned} C &= K(p \ a \ y) \bmod 26 \\ &= K(15 \ 0 \ 24) \bmod 26 \\ &= (375 \ 819 \ 486) \bmod 26 \\ &= (11 \ 13 \ 18) = LNS \end{aligned}$$

Hill密码

- Hill密码解密需要用到K矩阵的逆 K^{-1} 。
 - $K^{-1} K = K K^{-1} = I$ (单位矩阵)
 - 并不一定所有的矩阵都有逆。
- 加密过程 $C = KP \bmod 26$
- 解密过程 $P = K^{-1} C \bmod 26$
- Hill的优点是完全隐蔽了单字母的频率特性。Hill用的矩阵越大，所隐藏的信息就越多
 - 例如 3×3 的矩阵就隐藏了双字母的频率特性

Hill密码练习

- 例：Hill密码的秘钥K如下示，请对明文“THU”加密

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
0	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

- $C = K(T \ H \ U) \bmod 26$
- $C = K(19 \ 7 \ 20) \bmod 26$
- $C = (542 \ 945 \ 432) \bmod 26$
- $C = (22 \ 9 \ 16) = WJQ$

$$K = \begin{pmatrix} 17 & 17 & 5 \\ 21 & 18 & 21 \\ 2 & 2 & 19 \end{pmatrix}$$

Vigenere密码

- Vigenere密码是使用一系列Casear密码组成密码字母表的加密算法，属于多表代换密码
- 多表代换密码的共同特性：
 - 采用相关的单表代换规则集
 - 由密钥决定给定变换的具体规则
- Vigenere密码的强度在于每个明文字母对应着多个密文字母，并且使用唯一的字母作为密钥词
 - 字母频率信息被隐藏了
 - 但并非所有的明文结构信息都被隐藏

Vigenere密码

- 为了生成密码，需要使用表格法
- 表格包括了26行字母表，每一行都由前一行向左偏移一位得到
- 具体使用哪一行字母表进行编译是基于密钥进行的，在过程中会不断地变换

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

Vigenere密码

- 设明文为：ATTACKATDAWN；密钥为：THUCS

明文	A	T	T	A	C	K	A	T	D	A	W	N
密钥	T	H	U	C	S	T	H	U	C	S	T	H
密文												
明文	A	T	T	A	C	K	A	T	D	A	W	N

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

Vernam密码和一次一密

- 1918年，工程师Gilbert Vernam引入了这样一种体制：选择一个与明文毫无统计关系并且和明文一样长的密钥，其运算是基于二进制数据而非字母的
 - $c_i = p_i \oplus k_i$
- 陆军情报局军官Joseph Mauborgne又提出了一种Vernam密码的改进方案：
 - 使用与消息一样长且无重复的随机密钥来加密消息，这就是著名的一次一密（One-time Pad; OTP），它是不可攻破的，即无条件安全的

Vernam密码和一次一密

- 在理论上，一次一密是牢不可破的
 - 1949年，香农(Claude Elwood Shannon) 证明了它的牢不可破
- 虽然一次一密在理论上的安全性无庸置疑，但在实际操作上却存在很大的困难：
 - 产生大规模随机密钥很困难：用以加密的文本，也就是一次性密码本，必须是无特定规律的
 - 它可以是一串随机数字，一句话，或者一本英文名著
 - 它必须至少比被加密的文件等长
 - 密钥的分配和保护很困难
 - 一次性密码本只能用一次，且必须对非关系人小心保密
 - 不再使用时，用以加密的文本应当要销毁，以防重复使用

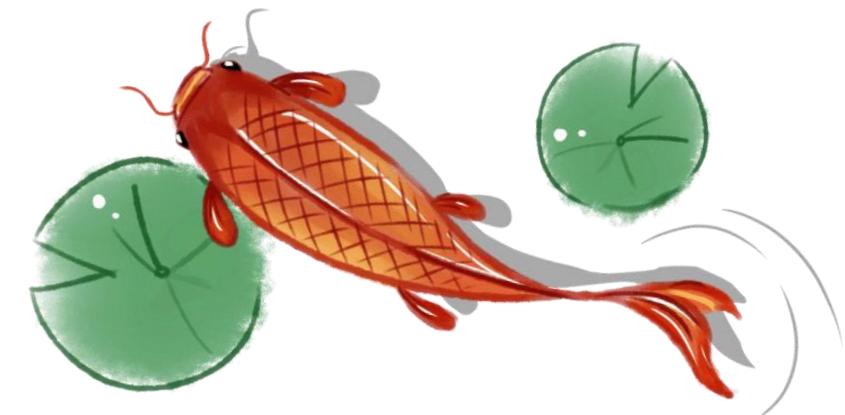
Vernam密码和一次一密

明文	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
数字	0	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

明文	T	H	I	S	I	S	A	N	E	X	A	M	P	L	E
明文数字	19	7	8	18	8	18	0	13	4	23	0	12	15	11	4
密钥	M	A	S	K	L	N	S	F	L	D	F	K	F	P	Q
密钥数字	12	0	18	10	11	13	18	5	11	3	5	10	9	15	16
+得密文数字	31	7	26	28	19	31	18	18	15	26	5	22	24	26	20
模26	5	7	0	2	19	5	18	18	15	0	5	22	24	0	20
密文	F	H	A	C	T	F	S	S	P	A	F	W	Y	A	U



古典密码[2]: 置换技术



置换技术Transposition Cipher

- 置换密码是通过置换形成新的排列
- 最简单的例子是栅栏技术
 - 按照对角线的顺序写入明文，按照行的顺序读出做为密文
- 更复杂的方案是把消息一行一行写成矩形块，然后按列读出，同时把列的次序打乱；列的次序就是算法的密钥
- 单步置换还是容易被识破，一般采用多步置换密码就安全多了

置换技术举例

- 明文：
 - Attack postponed until two am
- 一次置换后密文：
 - TTNAAPPTMTSUOAODWCOIXKN
LYPETZ
- 二次置换后密文：
 - NSCYAUOPTTWLTMDNAOIEPAX
TTOKZ
- 经过多次置换后，已没有什么规律可寻了

密钥	4	3	1	2	5	6	7
明文	a	t	t	a	c	k	p
	o	s	t	p	o	n	e
	d	u	n	t	i	l	t
	w	o	a	m	x	y	z

密钥	4	3	1	2	5	6	7
明文	t	t	n	a	a	p	t
	m	t	s	u	o	a	o
	d	w	c	o	i	x	k
	n	l	y	p	e	t	z

转轮机

- 转轮机采用多层加密原理

- 包括一组相互独立的旋转轮，电脉冲可以通过；每个圆筒有26个输入引脚和26个输出引脚，并且一一相连；每个圆筒就定义了一个单表代换，多个圆筒就是多表代换
- 1919年，德国工程师Arthur Scherbius发明了转轮密码机ENIGMA；1944年，4轮ENIGMA装备了德国海军
- 英国的TYPEX打字密码机，是德国3轮ENIGMA的改进型，在英国通信中使用广泛



SPY Museum



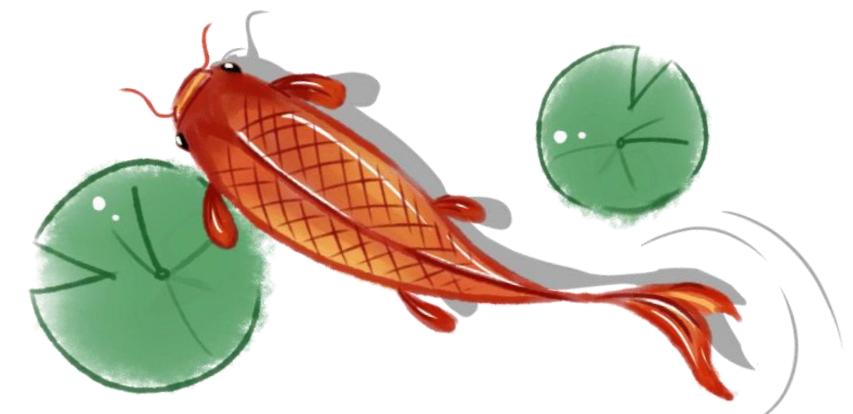
One-time Pad (Silk)





古典密码[3]：破译举例

- 频率分析法破译单表代替密码



单表代换密码的破译

- 基本方法有“穷举法”和“频率分析法”

- 穷举法

- 简单地一个接一个地去试每种可能的密钥，并且检查所得的明文是否有意义，就可分析出其密码系统
 - 也叫“蛮力攻击”，但不能用于一次一密

- 频率分析法

- 充分利用字符的统计特性和语言知识，并结合“猜字”的一种破译方法

频率分析法

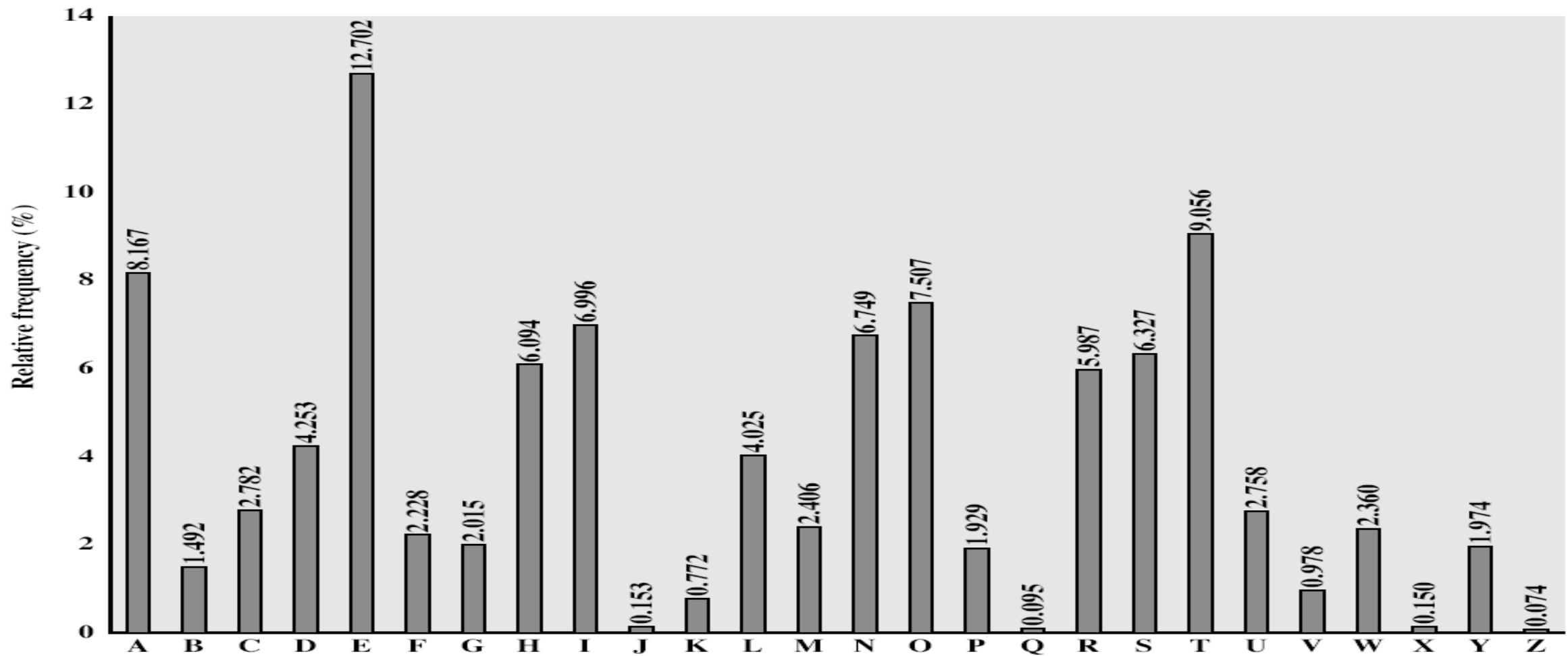
- 频率分析法就是充分利用语言的统计规律进行破译的
- 任何语言都有它自己的统计特性
 - 在英语中，最常出现的字母是e，字母q后面总是跟着字母u，等等
- 对于英语，常用的统计特性是单码统计特性、双码统计特性，有的时候还有多码统计特性
 - 下面以英语的统计特性为例，分析单表代换密码的破译

语言的统计特征：以英语为例

字母	概率	字母	概率	字母	概率
A	8.17%	J	0.15%	S	6.33%
B	1.49%	K	0.77%	T	9.06%
C	2.78%	L	4.03%	U	2.76%
D	4.25%	M	2.41%	V	0.98%
E	12.70%	N	6.75%	W	2.36%
F	2.28%	O	7.57%	X	0.15%
G	2.02%	P	1.93%	Y	1.97%
H	6.09%	Q	0.10%	Z	0.07%
I	6.97%	R	5.50%		

英语的一阶统计特性

- 极大概率字母：e



英语的二阶统计特性

- 根据Baker在1982年对100360字符样本的统计：
单码统计特性只反映了单个字符的统计特性，没有反映字符间的关系
- 英文字符的双码概率最大的十组字母依次为：
 - th he in er an re ed on es st
- 英文字符的三码概率最大的五组字母依次为：
 - the ing and her ere
 - the出现的概率大约是ing的三倍
 - 英文中大约有一半的单词以t,a,s和w开头
 - 英文中大约有一半的单词以e,s,d和t结尾

频率分析法的基本流程

- 做出单码频次统计表
- 利用以上信息，结合双码和三码统计特性，推断出一些可能的明密对应关系
- 利用语言的字符结合规律验证上述对应关系，并结合语言知识和猜字法推断出一些新的明密对应关系

举例：使用频率分析法破译下面的密文

- 截获的密文是：

UZQSOVUOHOXMOPVGPOZPEVSGZWSZOPFPESXUDBMET
SXAIZVUEPHZHMDZSHZOWSFAPPDTSVPQUZWYMXUZU
HSXE PYEPOP DZSZUF POMBZWPFUPZHMDJUDTMOHM

U	Z	Q	S	O	V	U	O	H	X	M	O	P	V	G	P	O	Z	P	E	V	S	G	Z	W	S
Z	O	P	F	P	E	S	X	U	D	B	M	E	T	S	X	A	I	Z	V	U	E	P	H	Z	H
M	D	Z	S	H	Z	O	W	S	F	P	A	P	P	D	T	S	V	P	Q	U	Z	W	Y	M	X
U	Z	U	H	S	X	E	P	Y	E	P	O	P	D	Z	S	Z	U	F	P	O	M	B	Z	W	P
F	U	P	Z	H	M	D	J	U	D	T	M	O	H	M											

使用单码统计特征

- 由频次分布表可看出，此段密文是单表代替加密的可能性很大，可以假设为单表代替密码体制，开始破译
- 出现频率最大的字符是P，很可能是字母e，当然Z也有可能是t

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
2	2	0	6	6	4	2	7	1	1	0	0	8	0	9	16	3	0	10	3	10	5	4	5	2	14

使用单码统计特征

- P → e ; Z → t

U	Z	Q	S	O	V	U	O	H	X	M	O	P	V	G	P	O	Z	P	E	V	S	G	Z	W	S
I	T	W	A	S							e			e		t	e						T	H	A
Z	O	P	F	P	E	S	X	U	D	B	M	E	T	S	X	A	I	Z	V	U	E	P	H	Z	H
T		e		e														t				e		t	
M	D	Z	S	H	Z	O	W	S	F	P	A	P	P	D	T	S	V	P	Q	U	Z	W	Y	M	X
		t		t		HA	VE	B	E	E	N						e	W	I	T	H				
U	Z	U	H	S	X	E	P	Y	E	P	O	P	D	Z	S	Z	U	F	P	O	M	B	Z	W	P
t					e				e		e		t		t		e						T	H	E
F	U	P	Z	H	M	D	J	U	D	T	M	O	H	M											
	e	t						I	N																

(1) ZWP极有可能是the

(3) WSFP极有可能是have

(5) 由UZ和UD猜U为i

(6) 进而确定Q为w, O为s

(2) ZWSZ极有可能是this或者that, 由于S在此文中出现了10次, 且a为较大概率字符, 故暂时将ZWSZ猜作that, (如果走不通, 再退回来重新试)

(4) 根据语法知识, 可以大胆确定密文APPD对应明文been

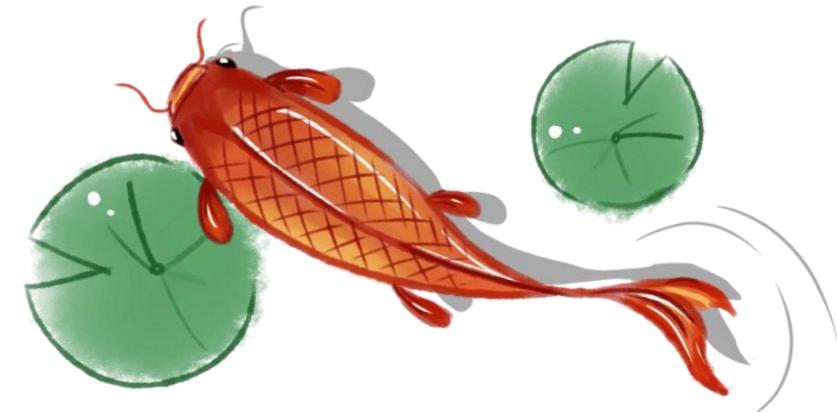
破译已经取得重大突破 下面继续用猜字法破译

- it was disclosed yesterday that several informal but direct contacts have been made with political representatives of the viet cong in moscow

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

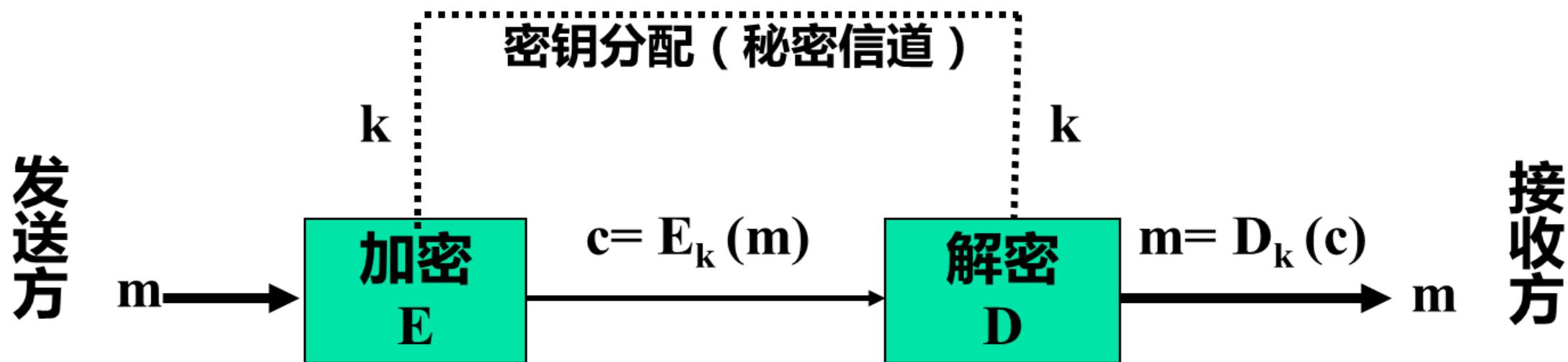


对称密码[1]: S-DES



对称密钥算法简介

- 加密和解密使用相同的密钥： $K_E = K_D$
- 密钥必须使用秘密的信道分配



对称密钥算法简介

- 常用对称密钥密码算法

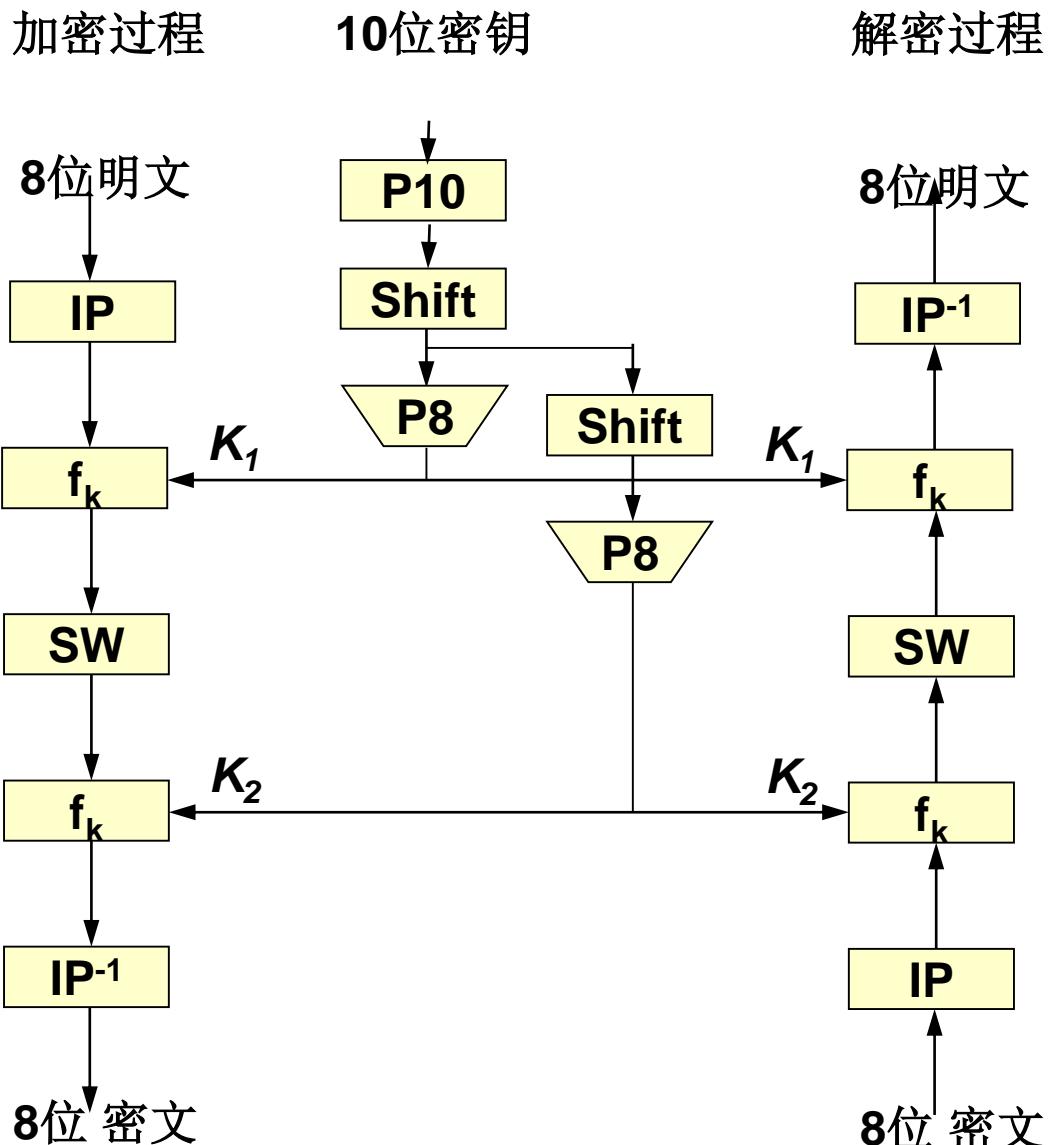
- DES (Data Encryption Standard)及其各种变形
- IDEA (International Data Encryption Algorithm)
- RC2, RC4, RC5,
- AES (Advanced Encryption Standard)
- CAST-128
- Blowfish

简化DES

- 简化DES(S-DES)是为教学使用的一个加密算法，与DES有着相似的性质和结构，但是参数要小很多，便于理解。
- S-DES加密算法的输入为一个8位明文组和一个10位的密钥，输出为8位的密文组。
- S-DES解密算法的输入为一个8位密文组和一个10位的密钥，输出为8位的明文组。

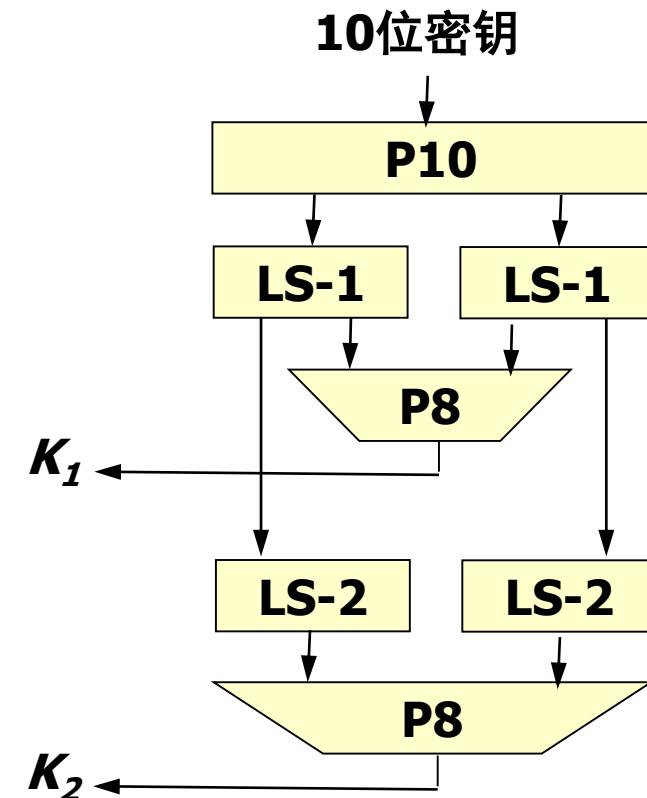
- 加密算法包括5个函数：
- 初始置换IP
- 复杂函数 f_k ，包含了置换和代换运算，并且依赖于密钥
- 用于转换数据两个部分的简单置换函数SW
- 再一次的复杂函数 f_k
- 置换函数IP的逆函数IP-1
- 密文= $IP^{-1}(fk_2(SW(fk_1(IP(\text{明文}))))))$
- 明文= $IP^{-1}(fk_1(SW(fk_2(IP(\text{密文}))))))$
- $K_1 = P8(\text{Shift}(P10(key)))$
- $K_2 = P8(\text{Shift}(\text{Shift}(P10(key))))$

S-DES的整体结构

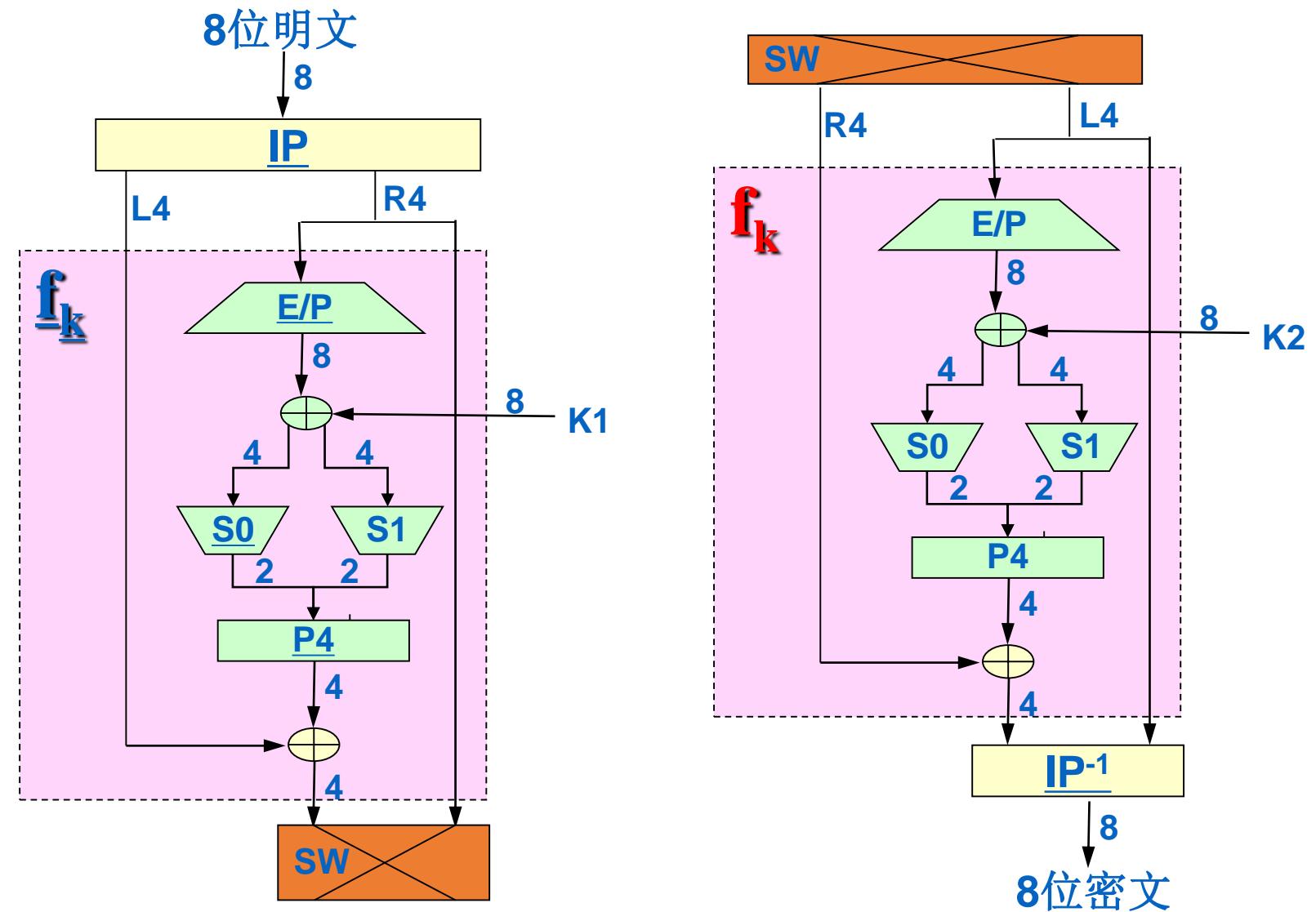
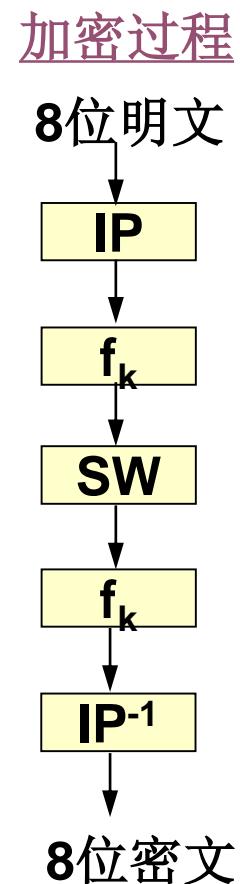


S-DES密钥的产生

- S-DES依赖于收发双方共享的10位密钥，它产生的两个8位子密钥分别用在加密和解密的不同阶段。
- 10位密钥即($k_1, k_2, k_3, k_4, k_5, k_6, k_7, k_8, k_9, k_{10}$)。
- 置换P10定义为
 $P10(k_1, k_2, k_3, k_4, k_5, k_6, k_7, k_8, k_9, k_{10}) = (k_3, k_5, k_2, k_7, k_4, k_{10}, k_1, k_9, k_8, k_6)$
- LS-1表示前5位和后5位分别循环左移1位。
- LS-2表示前5位和后5位分别循环左移2位。
- 置换P8定义为
 $P8(k_1, k_2, k_3, k_4, k_5, k_6, k_7, k_8, k_9, k_{10}) = (k_6, k_3, k_7, k_4, k_8, k_5, k_{10}, k_9)$
- 由此可以计算得到子密钥K1和K2。



S-DES加密过程



S-DES加密

- 加密过程包括5个函数
- 初始置换IP
 - $IP(n_1, n_2, n_3, n_4, n_5, n_6, n_7, n_8) = (n_2, n_6, n_3, n_1, n_4, n_8, n_5, n_7)$
- $E/P(n_1, n_2, n_3, n_4) = (n_4, n_1, n_2, n_3, n_2, n_3, n_4, n_1)$
- 将8位子密钥K1或K2与之进行按位异或。
- 前面4位(第一行)输入到S盒S0中得到一个2位输出，后面4位(第二行)输入到S盒S1中得到另外一个2位输出。

S-DES加密：映射F

- S盒的操作如下：

- 第1、4位作为二进制数决定S盒的行
- 第2、3位作为二进制数决定S盒的列
- 输出即是二进制的2位输出

$$S0 = \begin{matrix} & 0 & 1 & 2 & 3 \\ 0 & 1 & 0 & 3 & 2 \\ 1 & 3 & 2 & 1 & 0 \\ 2 & 0 & 2 & 1 & 3 \\ 3 & 3 & 1 & 3 & 2 \end{matrix}$$

- 例如：对S0盒输入(0100)

- 计算行(P_1, P_4)=(00);
计算列 (P_2, P_3)=(10)
- 输出是S0盒第0行第2列的数3，
即二进制 (11)

$$S1 = \begin{matrix} & 0 & 1 & 2 & 3 \\ 0 & 0 & 1 & 2 & 3 \\ 1 & 2 & 0 & 1 & 3 \\ 2 & 3 & 0 & 1 & 0 \\ 3 & 2 & 1 & 0 & 3 \end{matrix}$$

S-DES加密：映射F

- 由S0和S1输出构成的4位再进行一次置换
 $P4(n_1, n_2, n_3, n_4) = (n_2, n_4, n_3, n_1)$
- $P4$ 输出和 $L4$ 按位异或， 和 $R4$ 组合后就是函数F的输出
- 交换函数SW
 - SW将输入的左4位和右4位交换
- 末尾置换IP-1
 - $IP-1(n_1, n_2, n_3, n_4, n_5, n_6, n_7, n_8) = (n_4, n_1, n_3, n_5, n_7, n_2, n_8, n_6)$

S-DES加密：复杂函数fk

- 复杂函数fk
- 设L和R分别是fk的8位输入的左边4位和右边4位。F是一个4位串到4位串的映射。则
 - $fk(L, R) = (L \oplus F(R, SK), R)$
 - 其中SK是子密钥。
 - 例：IP置换的输出为(1011 1101)， $F(1101, SK) = (1110)$ ，求fk。则
 $L \oplus F(R, SK) = (1011) \oplus (1110) = (0101)$ ，所以 $fk(1011 1101) = (0101 1101)$ 。

S-DES实例

- 使用S-DES，
用密钥(01 11 11 11 01)手工加密二进制串 (1010 0010)

- $P_{10}(k_1, k_2, k_3, k_4, k_5, k_6, k_7, k_8, k_9, k_{10}) = (k_3, k_5, k_2, k_7, k_4, k_{10}, k_1, k_9, k_8, k_6)$

- $P_8(k_1, k_2, k_3, k_4, k_5, k_6, k_7, k_8, k_9, k_{10}) = (k_6, k_3, k_7, k_4, k_8, k_5, k_{10}, k_9)$

- $IP(n_1, n_2, n_3, n_4, n_5, n_6, n_7, n_8) = (n_2, n_6, n_3, n_1, n_4, n_8, n_5, n_7)$

- $IP^{-1}(n_1, n_2, n_3, n_4, n_5, n_6, n_7, n_8) = (n_4, n_1, n_3, n_5, n_7, n_2, n_8, n_6)$

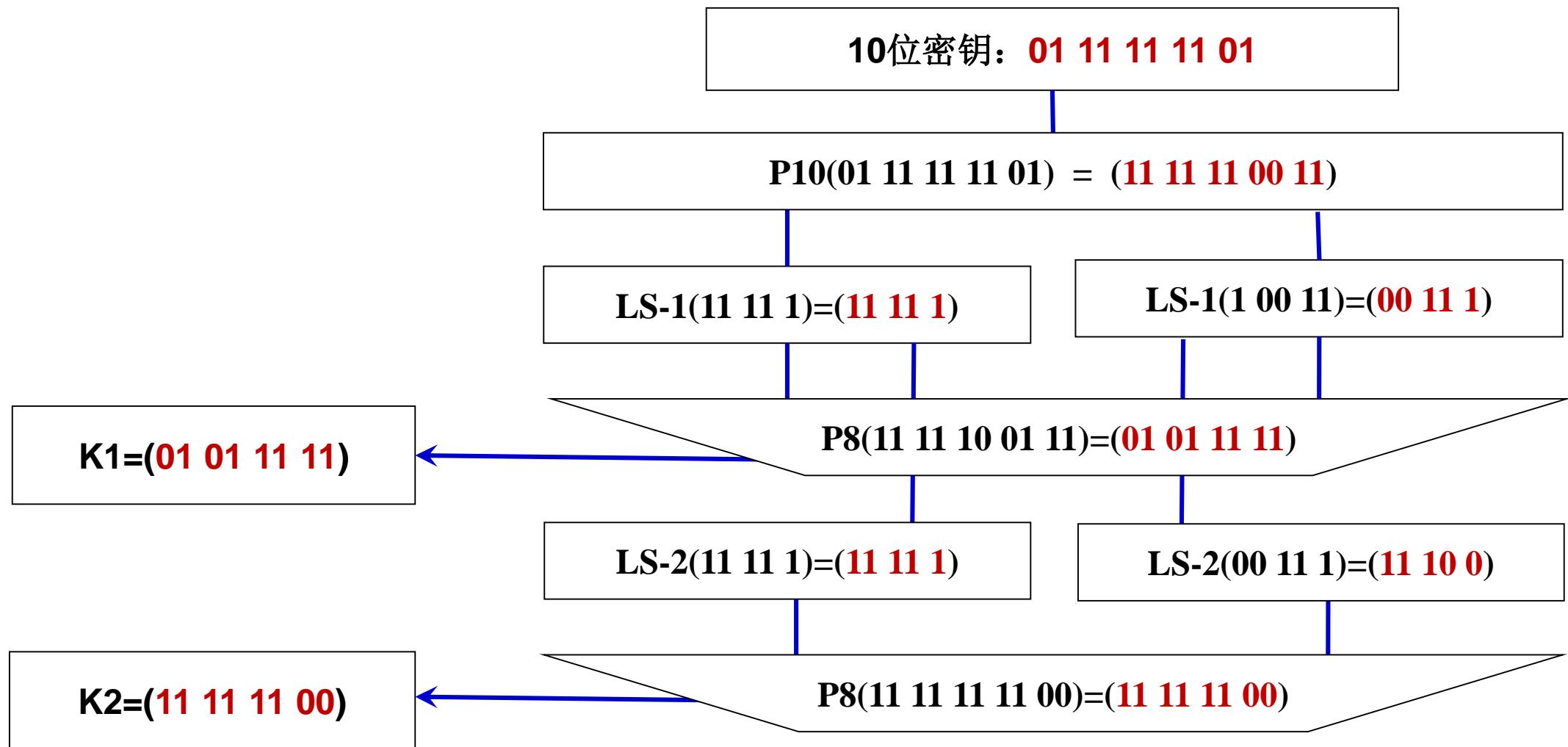
- $E/P(n_1, n_2, n_3, n_4) = (n_4, n_1, n_2, n_3, n_2, n_3, n_4, n_1)$

- $P_4(n_1, n_2, n_3, n_4) = (n_2, n_4, n_3, n_1)$

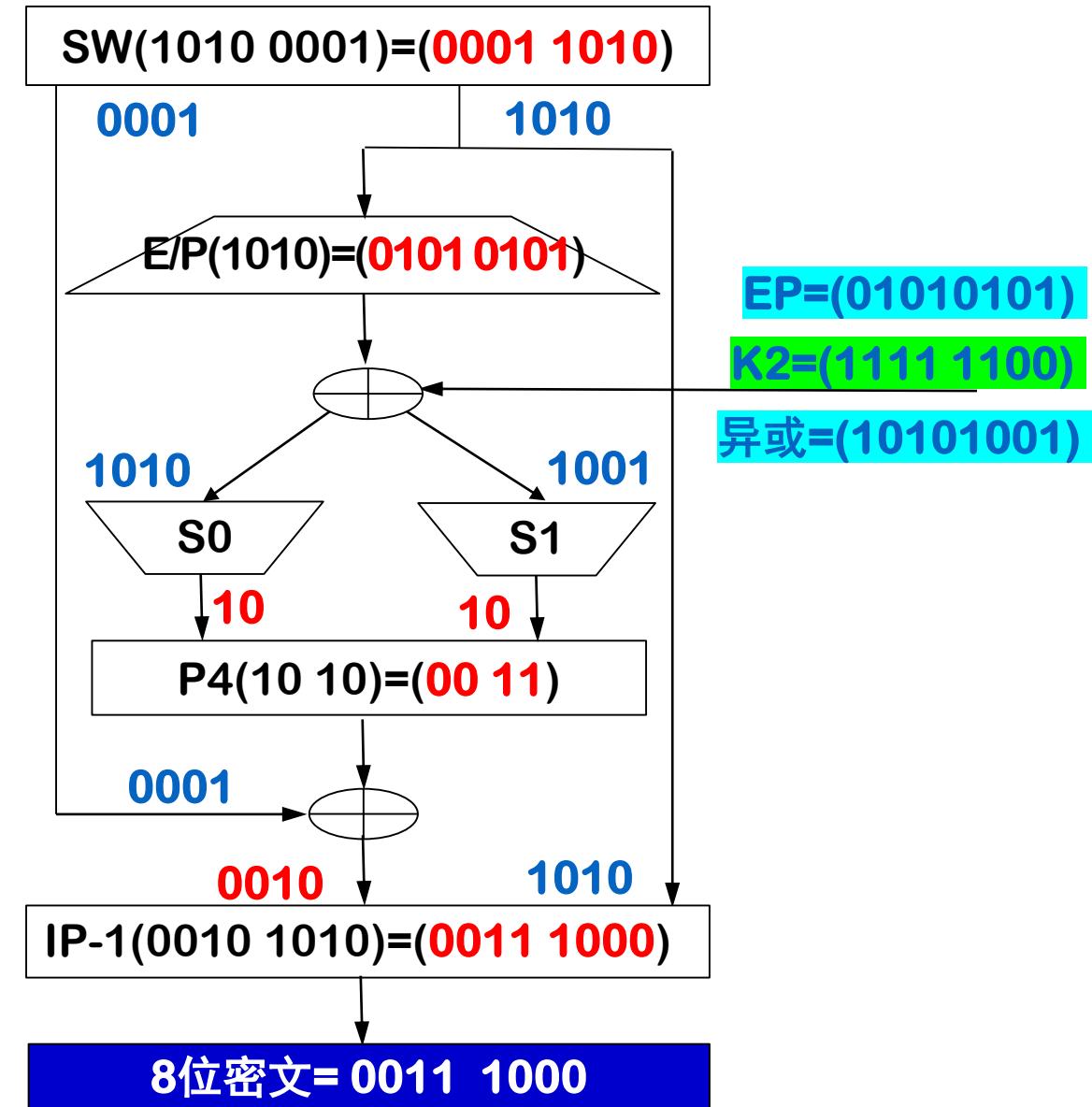
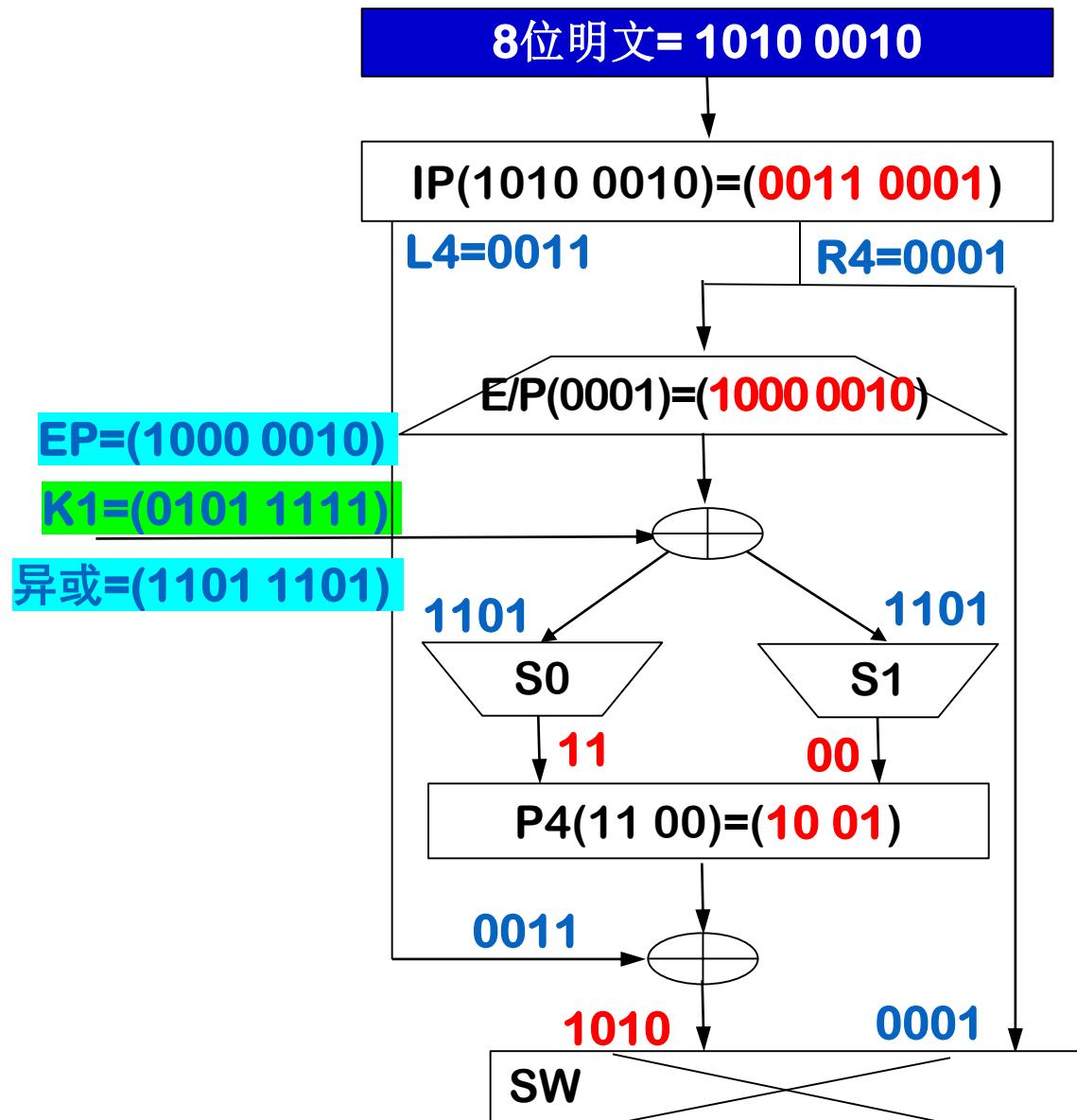
$$S_0 = \begin{matrix} & \begin{matrix} 0 & 1 & 2 & 3 \end{matrix} \\ \begin{matrix} 0 \\ 1 \\ 2 \\ 3 \end{matrix} & \left[\begin{array}{cccc} 1 & 0 & 3 & 2 \\ 3 & 2 & 1 & 0 \\ 0 & 2 & 1 & 3 \\ 3 & 1 & 3 & 2 \end{array} \right] \end{matrix}$$

$$S_1 = \begin{matrix} & \begin{matrix} 0 & 1 & 2 & 3 \end{matrix} \\ \begin{matrix} 0 \\ 1 \\ 2 \\ 3 \end{matrix} & \left[\begin{array}{cccc} 0 & 1 & 2 & 3 \\ 2 & 0 & 1 & 3 \\ 3 & 0 & 1 & 0 \\ 2 & 1 & 0 & 3 \end{array} \right] \end{matrix}$$

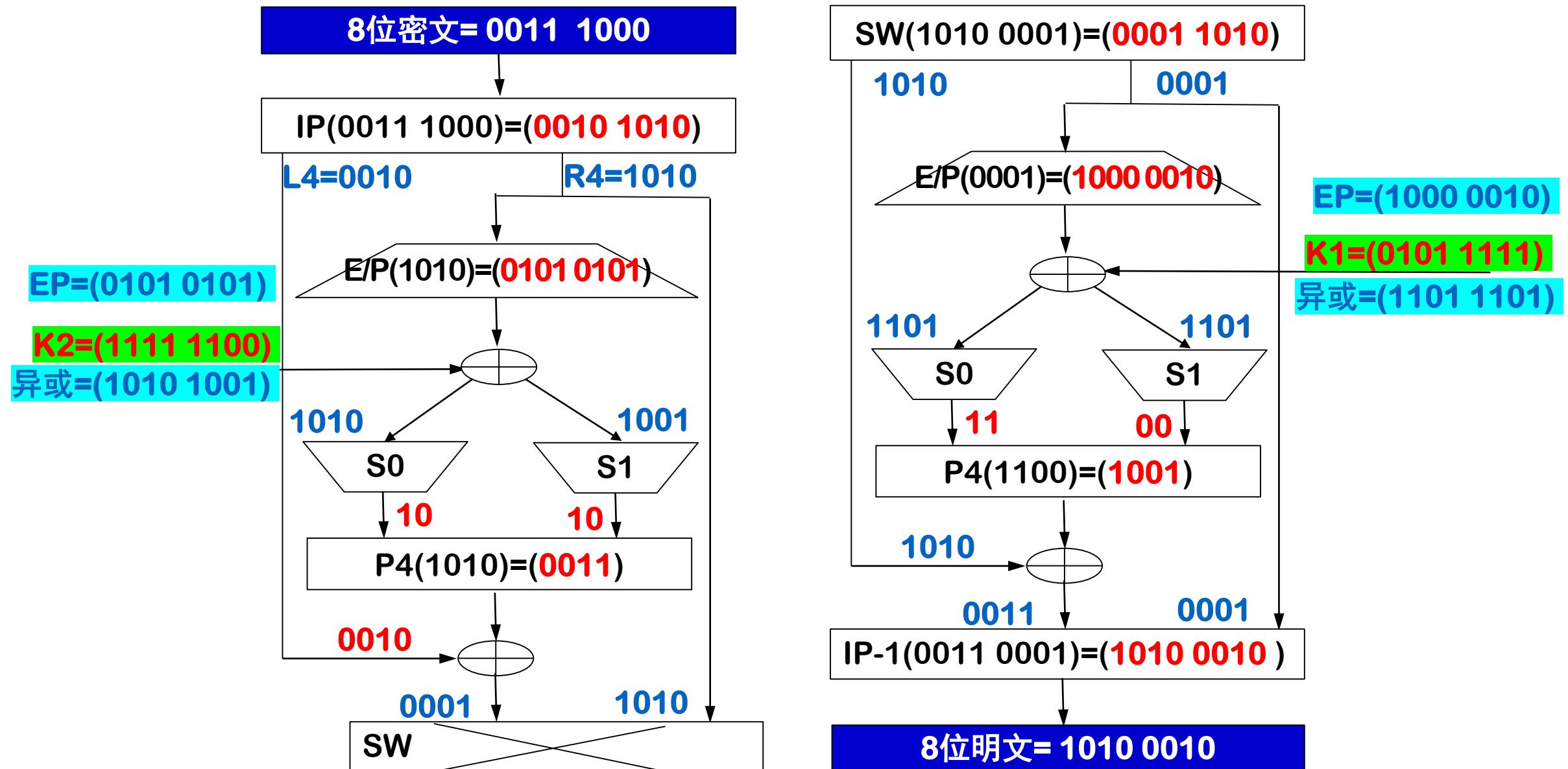
S-DES实例：计算子密钥



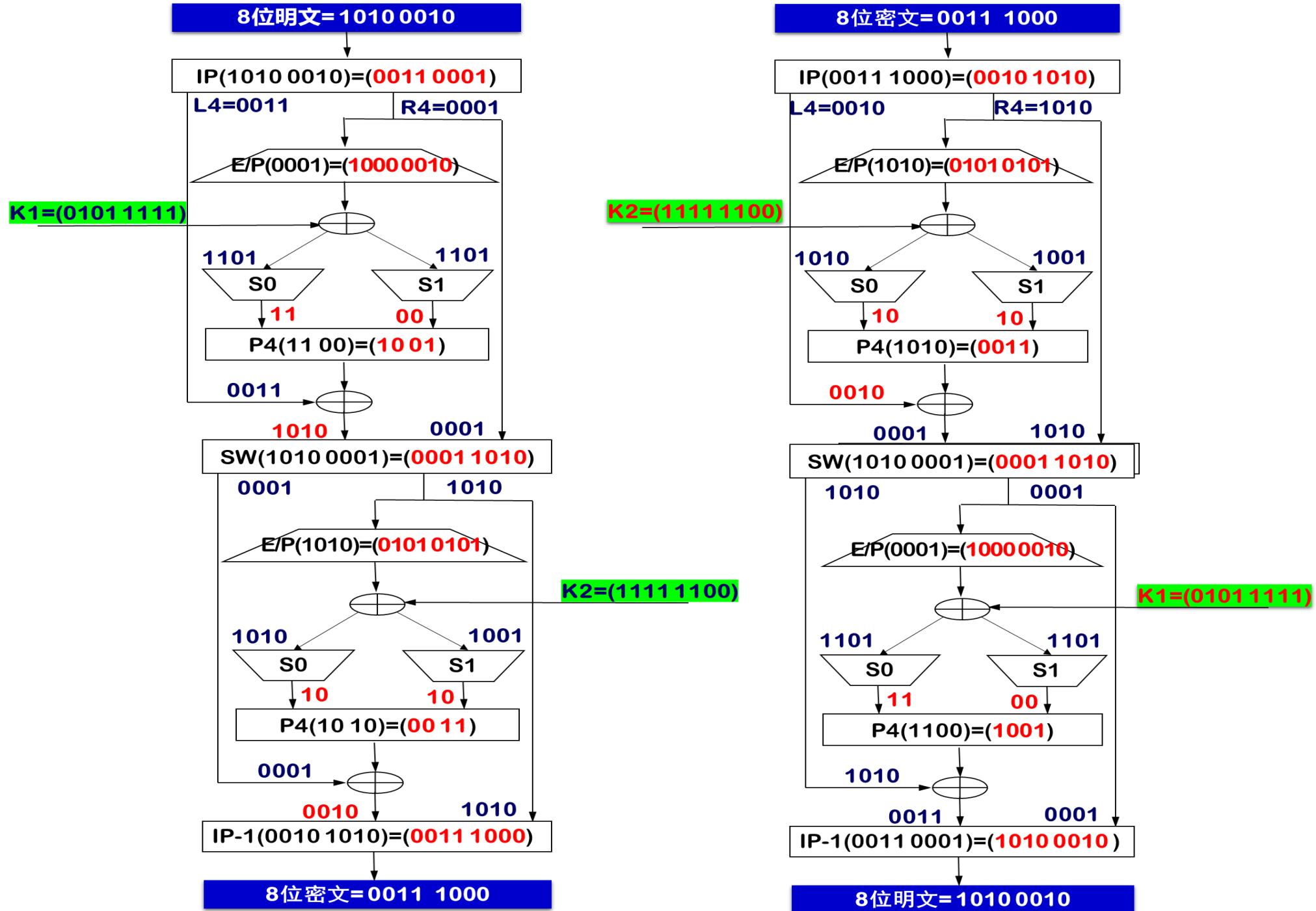
S-DES实例：加密过程



S-DES实例：解密过程

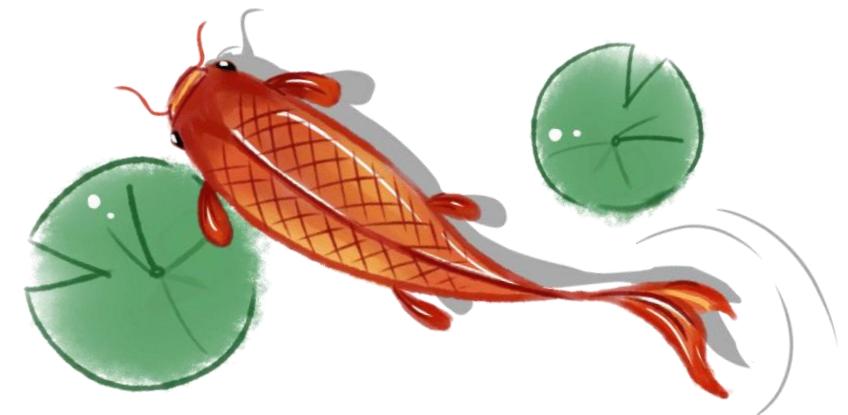


认真观察





对称密码[2]：Feistel密码结构



流密码和分组密码

- 流密码每次加密数据流的一位或一个字节
 - Vigenere密码和Verman密码都是流密码
- 分组密码是将一个明文组作为整体加密并且得到的是与之等长的密文组
 - 典型的分组大小是64bits或者128bits
 - 绝大多数基于网络的对称密码应用使用的都是分组密码
- 现在使用的对称分组密码算法都基于Feistel分组密码结构的

Feistel Cipher Structure

- Feistel建议使用乘积密码的概念来逼近简单代换密码
- 乘积密码是指依次使用两个或以上的基本密码，所得密码强度将强于所有单个密码的强度
- Feistel 建议交替使用代换和置换
 - 这实际上是Claude Shannon提出的交替使用混淆(Confusion)和扩散(Diffusion)乘积密码的实际应用
- 几乎所有的传统的分组密码算法的结构都有和 Feistel Cipher类似的结构

混淆和扩散

- Shannon引入混淆和扩展来刻画任何密码系统的两个基本构件，所关注的是如何挫败基于统计方法密码分析
- **扩散**是指使明文的统计特征消散在密文中，让每个明文数字尽可能地影响多个密文数字
- **混淆**是尽可能地使密文和加密密钥间的统计关系更复杂，以挫败推导出密钥的企图
- 混淆和扩散正是抓住了设计分组密码的本质而成为现代分组密码设计的里程碑

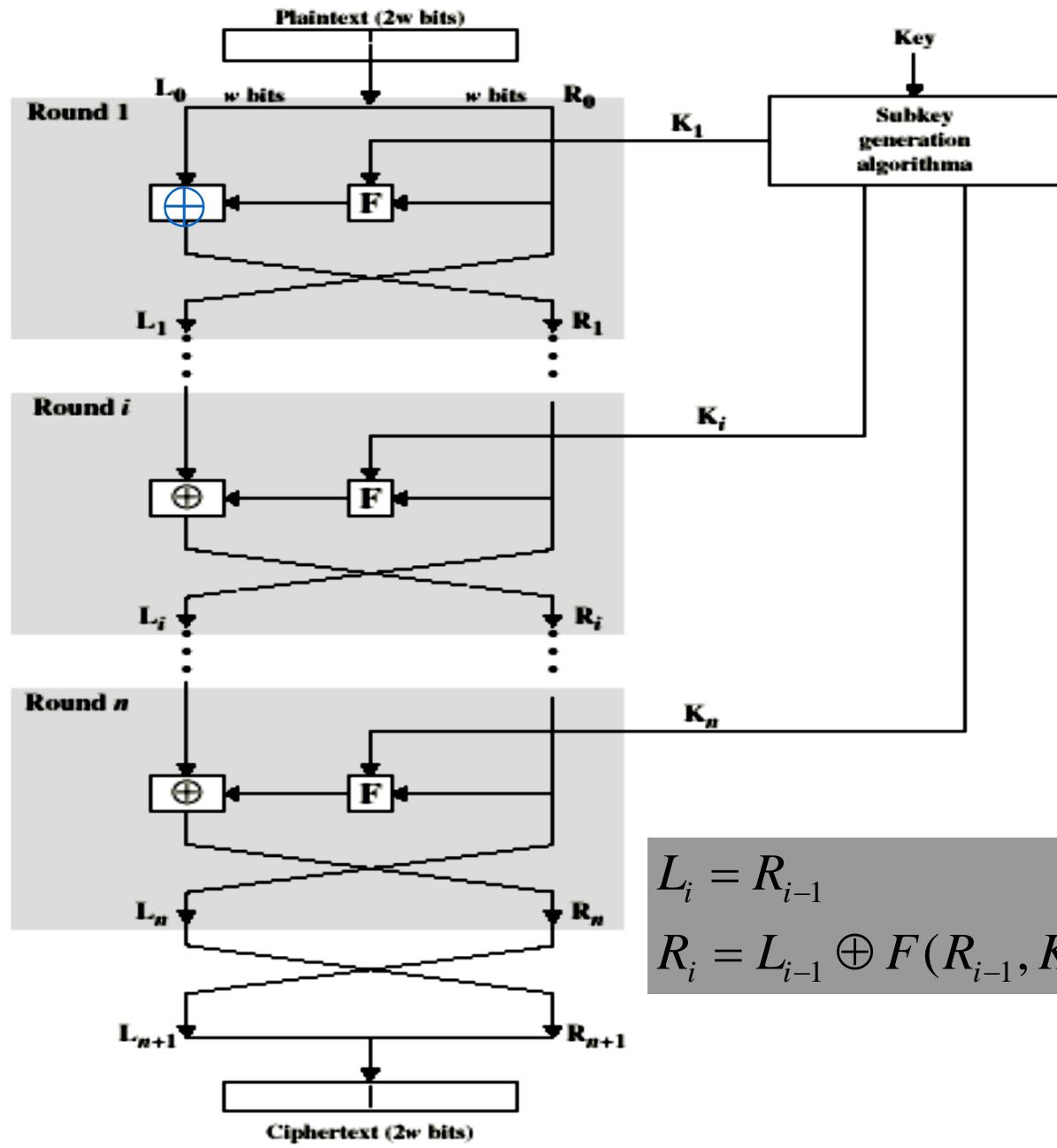


Figure 2.2 Classical Feistel Network

Feistel密码结构

- 输入是长度为2w位的明文组和密钥K
- 明文组被分成两个部分：L₀和R₀；这两半数据经过n轮迭代后组合成密文组
- 第i轮迭代的输入来自于上轮迭代的输出
- 子密钥K_i是由整个密钥K推导出来的
 - K_i不同于K，也互不相同

Feistel密码结构

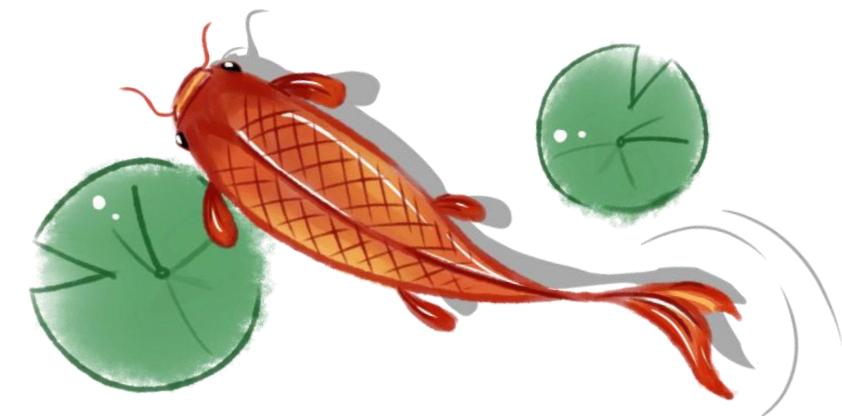
- 每轮迭代都有相同的结构
 - 代换作用在数据左半部分，它通过轮函数F作用在数据的右半部分后，与左半部分进行异或来完成
 - 每轮迭代的轮函数相同，但是输入的子密钥Ki不同
 - 代换之后，交换数据的左右两半完成置换

影响Feistel具体实现的重要参数

- 分组长度和密钥长度
 - 越长安全性越高，但是会降低加密解密速度
- 迭代轮数
 - 多轮加密可以取得高安全性
- 子密钥产生算法
 - 子密钥产生越复杂，密码分析攻击就越困难
- 轮函数
 - 轮函数越复杂，抗攻击能力越强
- 实现中还需考虑快速软件加解密和简化分析难度等问题



对称密码[3]: DES算法



DES的诞生

- 20世纪60年代后期，IBM公司成立了一个由Horst Feistel负责的计算机密码编码学研究项目
- 1971年设计出算法LUCIFER：分组长度为64位，密钥长度为128位，具有Feistel结构的分组密码
- 因为LUCIFER非常成功，IBM决定开发一个适合于芯片实现的商业密码产品
- 由Walter Tuchman和Carl Meyer负责，开发了一个抗分析攻击能力更强，密钥长度为56位的密码，这就是DES
- 1977年，DES被美国国家标准化局接收为国家密码标准方案

DES算法原理

- DES采用分组加密，是一种对称密钥算法
 - 密钥长度为56bits
(加上奇偶校验，通常写成64bits)
 - 分组长度是64 bits
- DES算法使用了标准的算术和逻辑运算

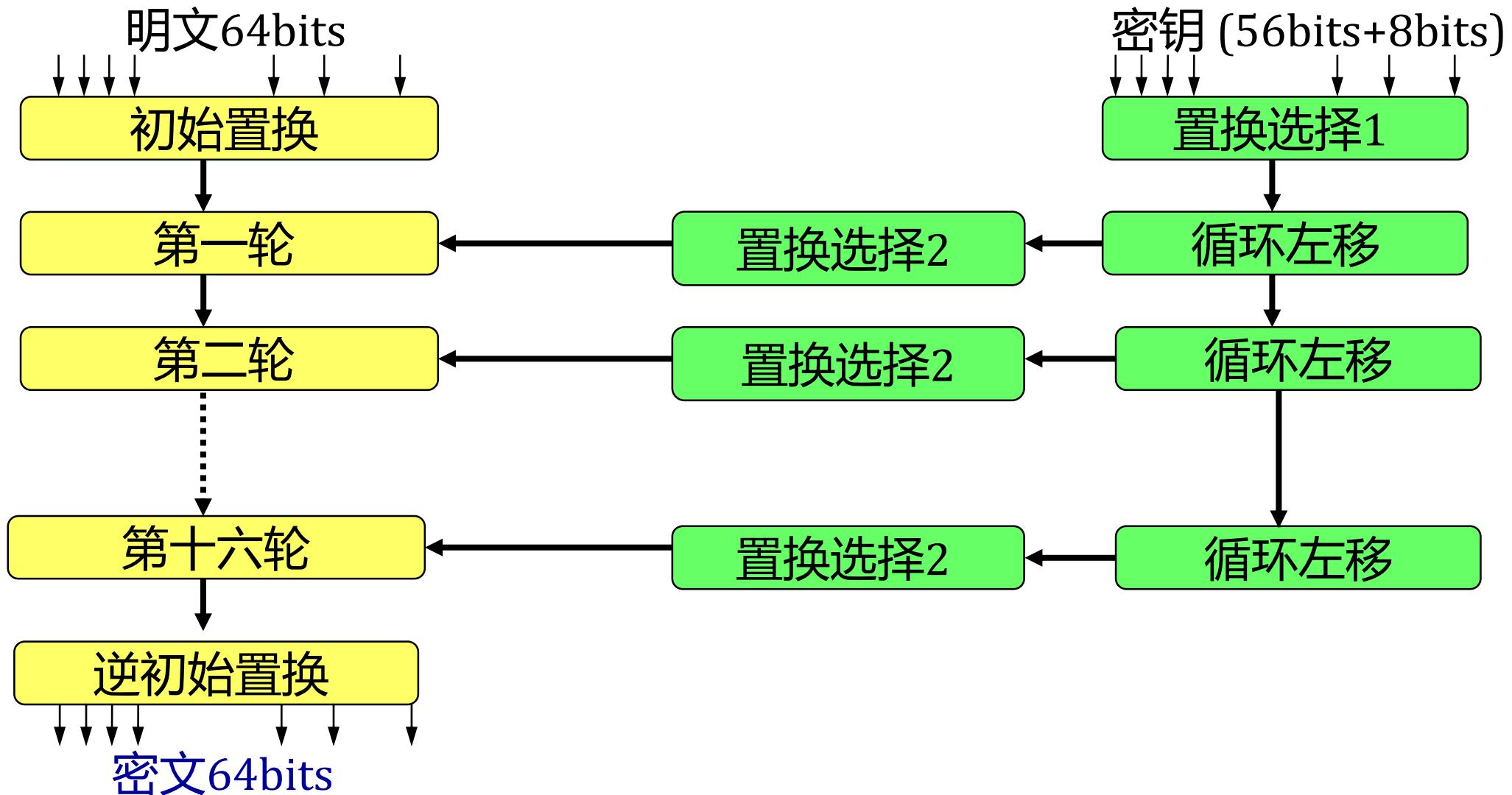
DES 加密过程

- 首先把明文分成以64 bit为单位的块m，对于每个m，执行如下操作

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

- 初始置换，IP
 - 16轮迭代， T_i , $i=1,2,\dots,16$
 - 末尾置换，IP-1
-
- 除了初始置换和末尾置换，DES的结构与Feistel密码结构完全相同

DES算法概要



子密钥生成

- 拆分：

- 56 bits 的密钥分成两部分， C_i , D_i ，各28bits

- 循环左移：

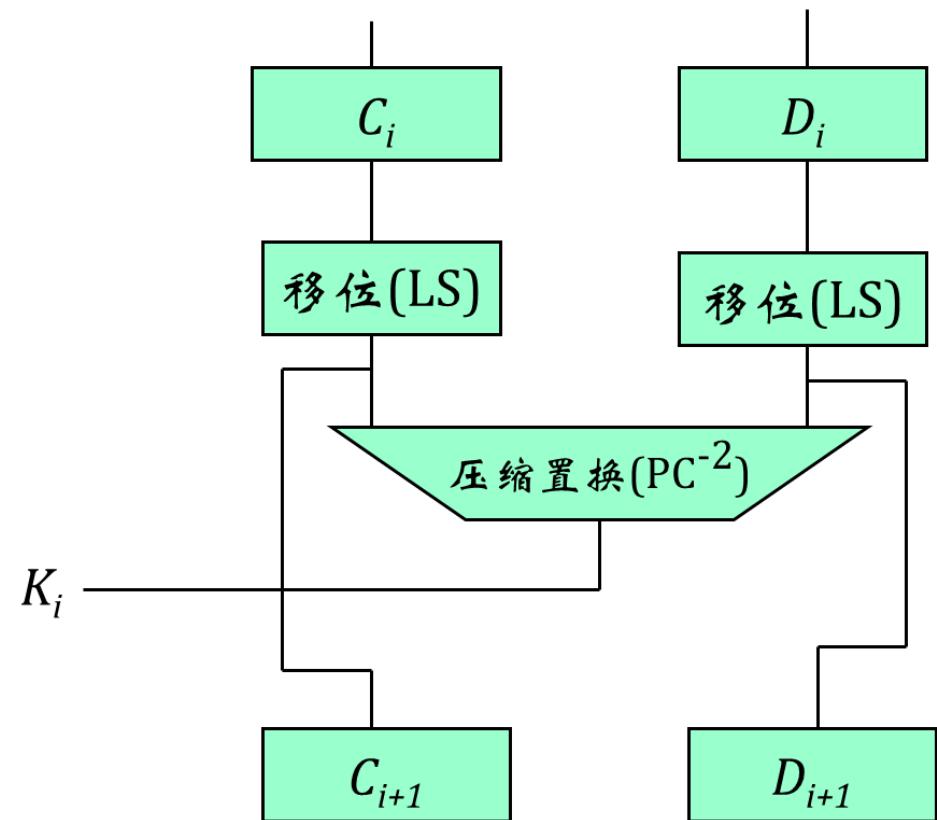
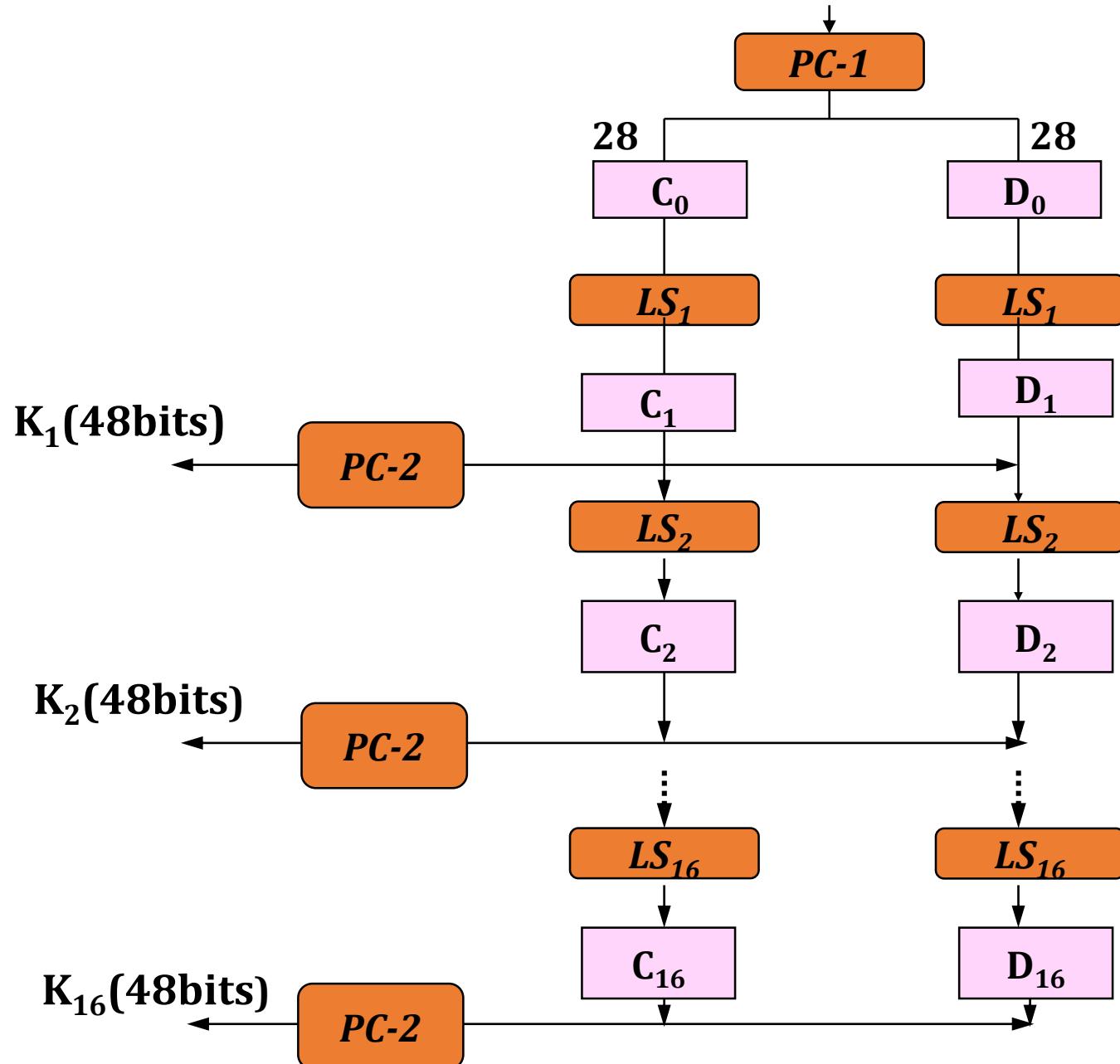
- 根据迭代的轮数，分别左移一位或两位

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

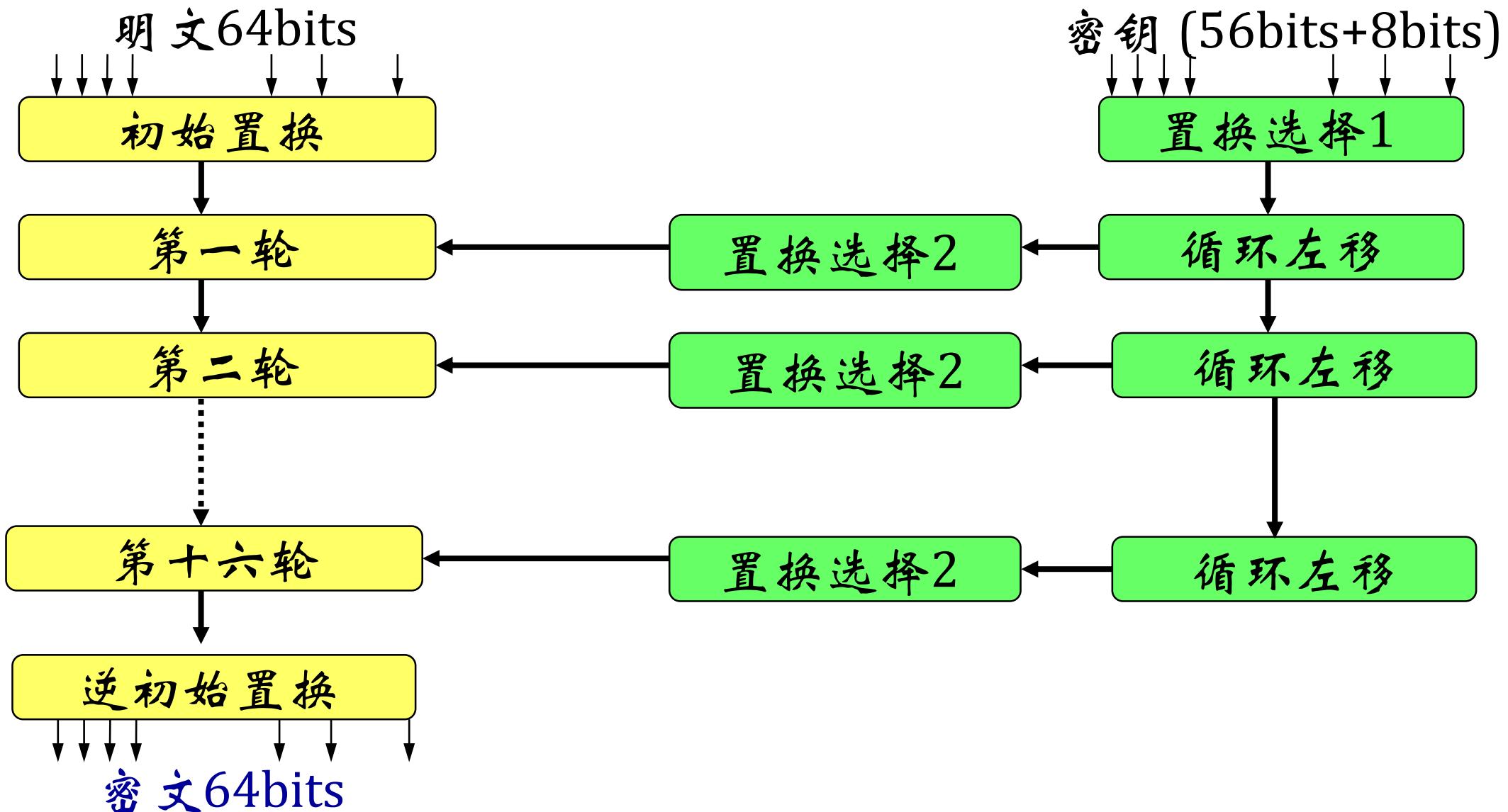
- 压缩置换 (PC-2)：从56bits中选择48bits

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				

子密钥的生成



DES算法概要

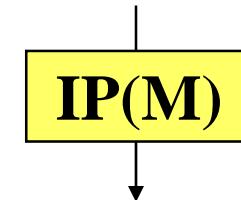


初始置换 (IP)

- 初始换位 (IP)

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

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



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

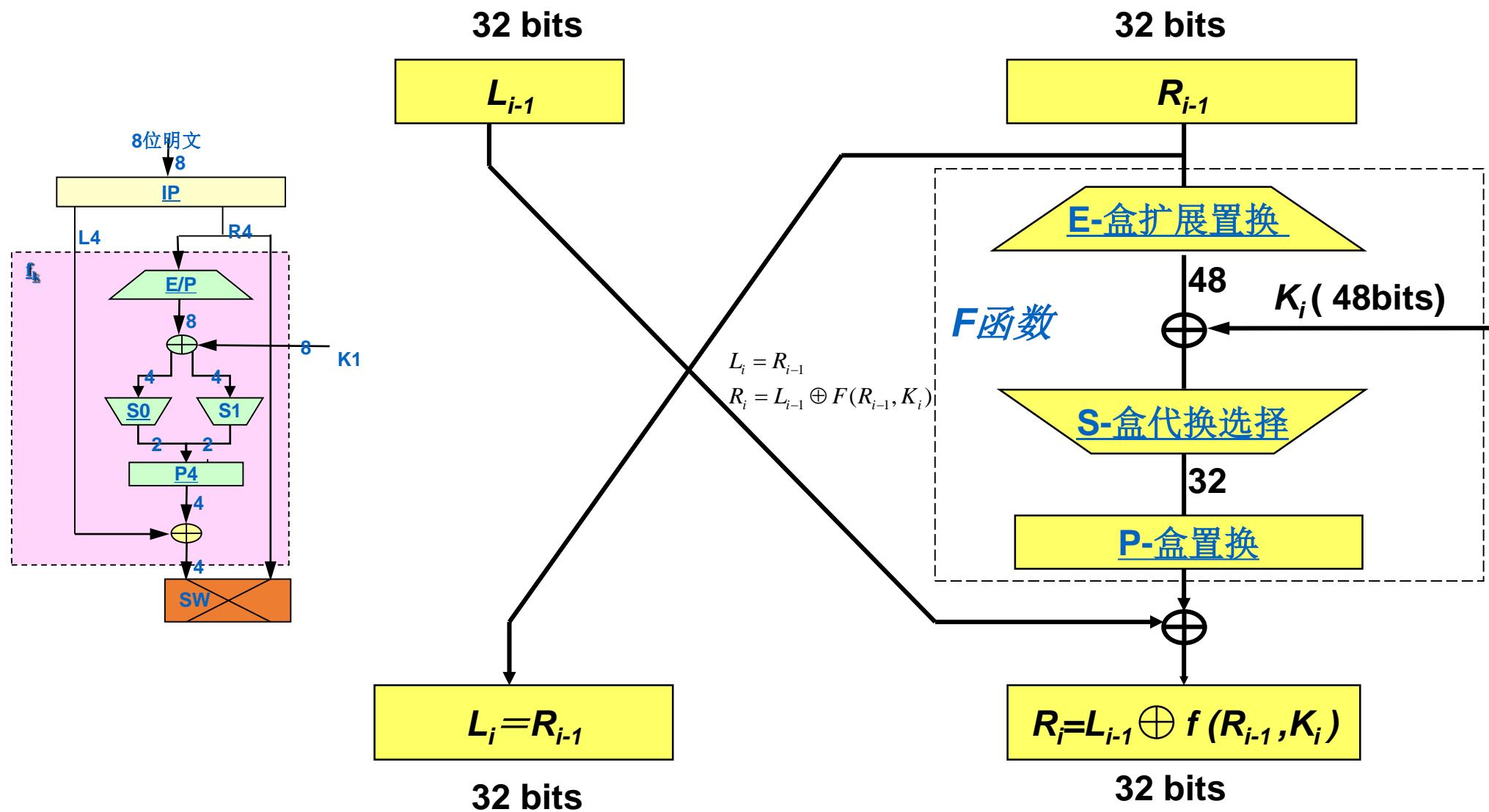
末尾置换 (IP-1)

- 末置换

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

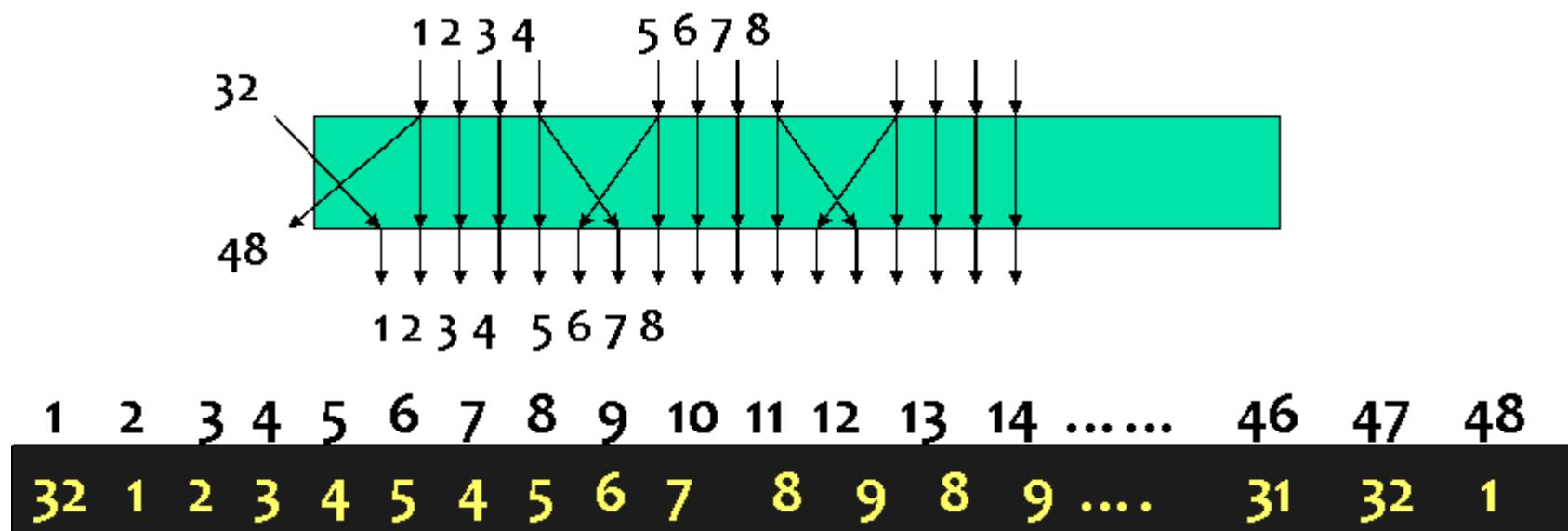
- $\text{IP-1}(\text{IP}(M))=M$

一轮迭代



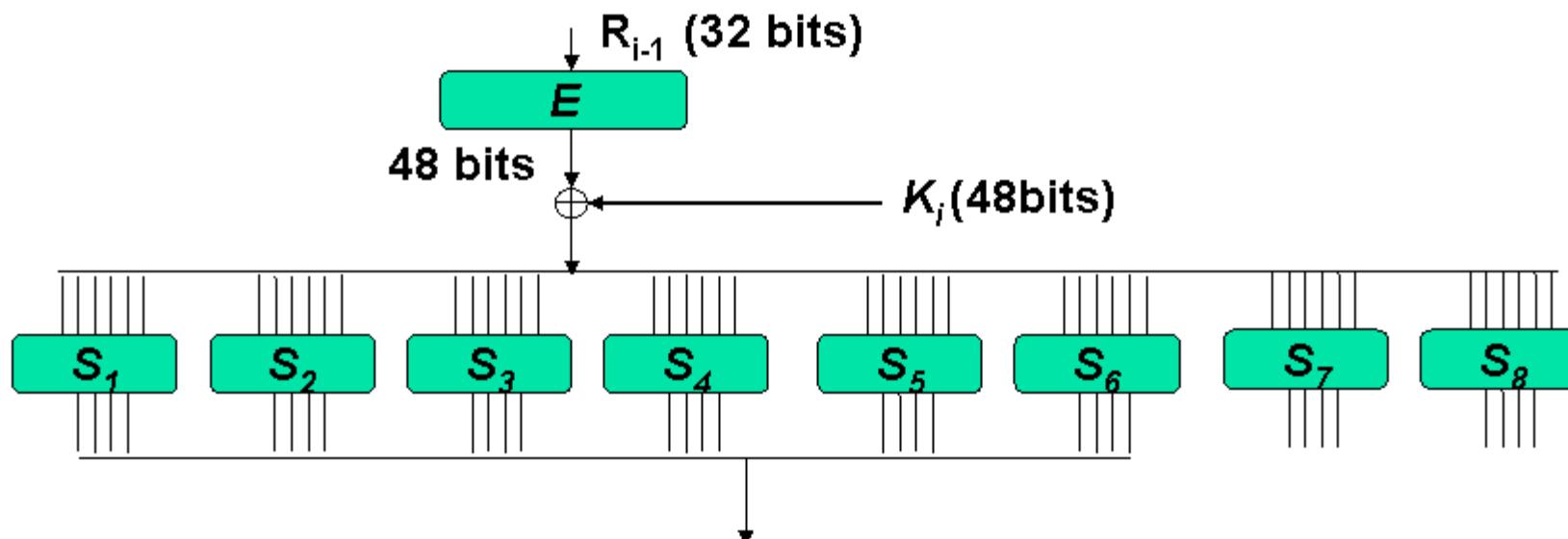
E盒扩展置换

- 将R_i从32位扩展到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



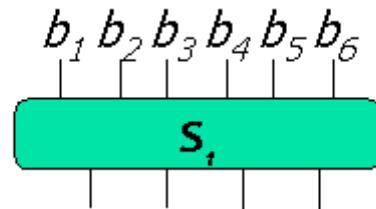
S-盒代换选择

- 将48比特压缩成32比特：代换函数由8个S盒组成，每个S盒都有6位输入产生4位输出
- 盒 S_i 输入(abcdef)，其中(af)决定了 S_i 盒中的行，(bcde)决定了 S_i 盒中列，取得 S_i 盒中一个元素N，将N转换成4位二进制即得到 S_i 盒输出



S-盒代换选择

- 输入6比特: $b_1 b_2 b_3 b_4 b_5 b_6$



- 输出4比特: $S(b_1 b_6, b_2 b_3 b_4 b_5)$

- 举例: $S_1(100110) = S_1(10, 0011) = S_1(2, 3) = 8 = (1000)$

		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
S_1	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
S_2	0	15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
	1	•	•
	2																
	3																

P-盒置换

- P-盒的输出：输入32bits，输出32bits。

1	2	3	4	5	6	7	8	9	30	31	32
16	7	20	21	29	12	28	17	1	15	.	.

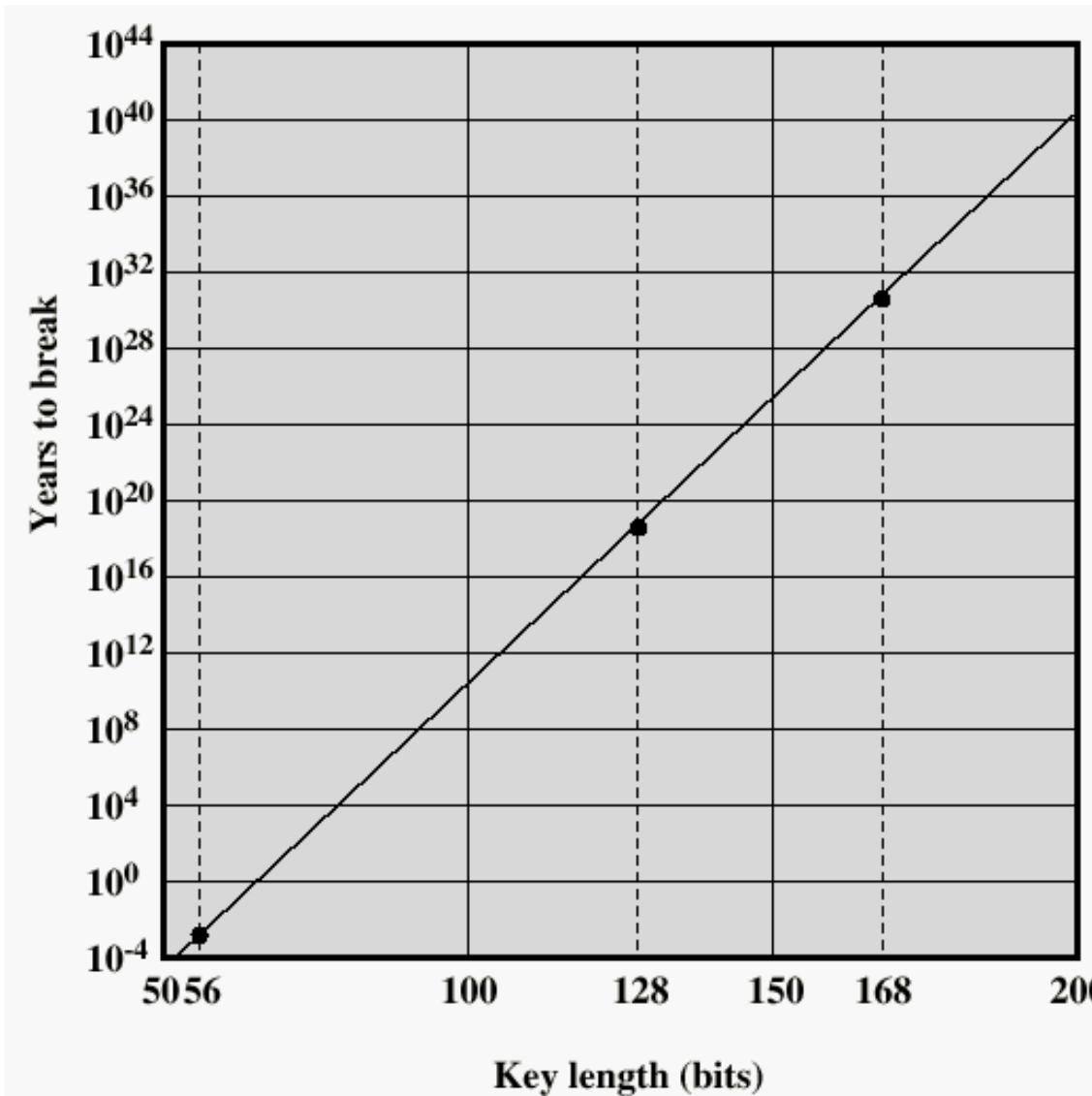
DES解密过程

- DES解密过程与加密过程完全相似，只不过将16次迭代的子密钥顺序倒过来，即

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

- 可以证明：
 - $DES(DES^{-1}(m))=m$

DES的安全性和速度



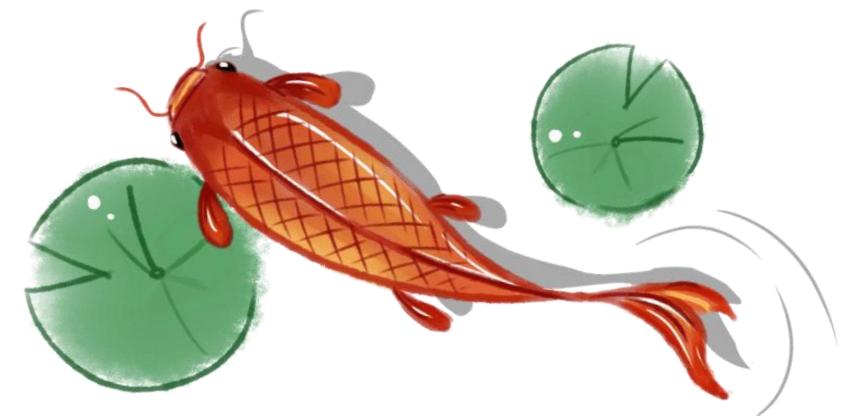
● 密钥的长度与安全性

- 1976年，耗资2000万美元的计算机，可以在一天中找到密钥。
- 1993年，100万美元的计算机，3.5小时用穷举法找到密钥。
- 现在，5000\$的计算机，可以在3.5小时找到密钥



对称密码[4]: 常用的对称密码

- 3-DES、Blowfish、RC5、AES

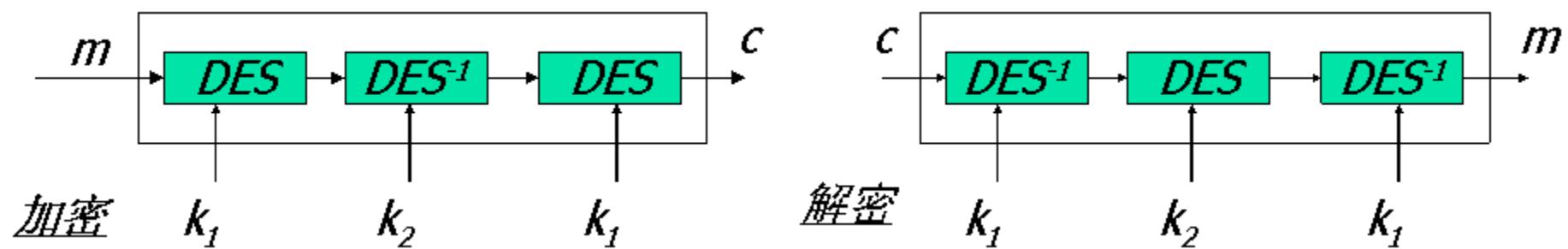


对称密码

- 当前最重要的几种对称密码算法，3DES、Blowfish和RC5，都具有以下几个标准：
 - 有相当高的密码强度
 - 广泛应用于Internet上面
 - 代表了自DES以来的对称密码技术
- 高级加密标准AES是一个对称分组密码算法，是美国国家标准技术局NIST2001年发布的，目的是用来取代DES。NIST预计在可以预见的未来3DES仍将是一个被认可的算法

3 DES

- 三重DES加密，密钥长度为112比特。已知明文攻击的代价上升到了 2^{112} 数量级，远远超出了人类现行的和未来的能力
- 两个密钥的三重DES是目前被广泛接收的加密算法，密钥 $k=k_1k_2$



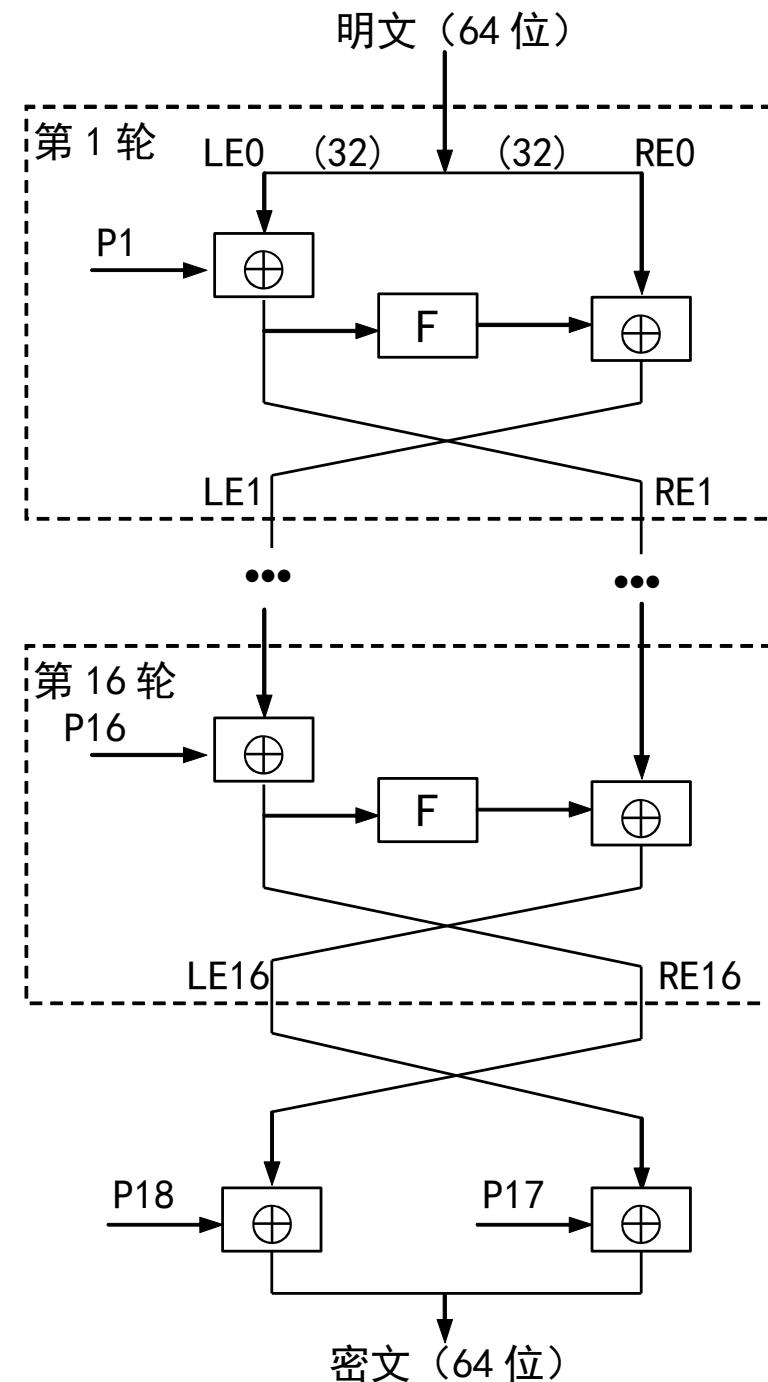
$$C = E_{K1}[D_{K2}[E_{K1}[P]]]$$

Blowfish算法

- Blowfish算法是由Bruce Schneier设计的一种对称分组密码；具有快速、紧凑、简单和安全性可变的特性
- Blowfish算法的分组长度是64位，密钥长度可以在32 ~ 448位之间变化
- Blowfish算法在很多产品上得到实现，其安全性至今没有受到挑战

Blowfish加密过程

- Blowfish算法使用两个基本的运算：
 - 加法：字的模 2^{23} 加法(+)
 - 按位异或：(\oplus)
- F映射中包含了四个S盒运算



Blowfish算法

- Blowfish算法的子密钥和S盒都是用Blowfish算法本身生成的，这使得数据完全不可以辨认，对它的密钥分析也就异常困难
- 与古典Feistel结构不同，Blowfish算法每轮运算都是对数据的左右两个部分同时执行运算，这使得密码的强度又增强了
- 因为密钥长度可以达到448位，完全可以抵抗穷举攻击

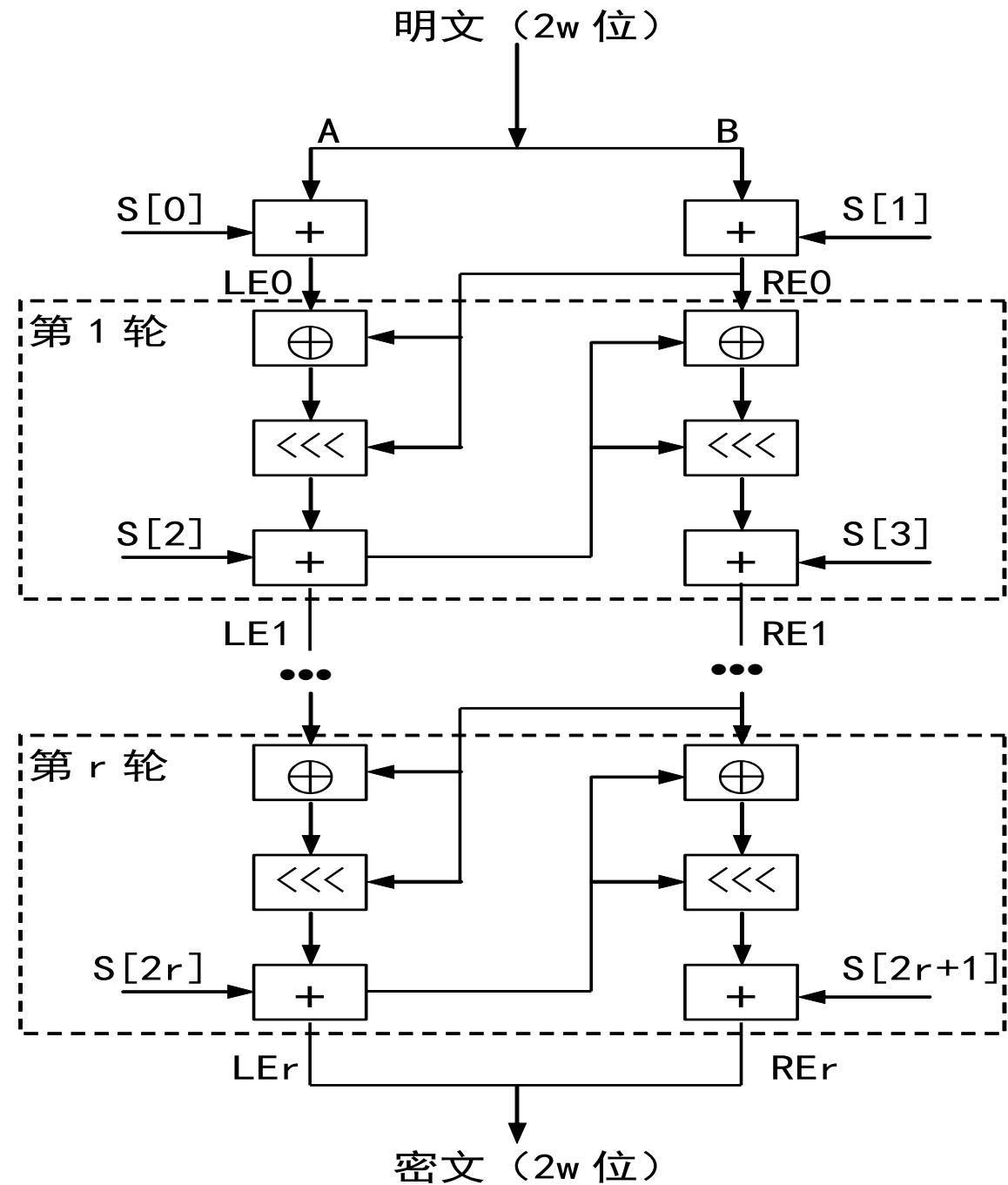
RC5

- RC5是Ron Rivest设计的一种对称加密算法；RC5适于软硬件实现，迭代次数和密钥长度可变、简单、安全性高
- RC5的分组长度可以位32、64或者128位，密钥长度则可取0~2040位
- RC5是一个由三个参数确定的加密算法族
 - w: 分组长度
 - r: 迭代次数
 - b: 密钥K的8位字节数

RC5加密算法

- RC5使用了下面三种运算：

- 加法：
字的模 2^w 加法(+)
- 按位异或：(\oplus)
- 循环左移：
字X循环左移(<<<)



AES的产生

- 如果仅考虑算法安全，3DES能够称为未来数十年的加密标准
 - 但是3DES最根本的缺点在于该算法用软件实现的速度慢
 - 而且3DES的分组长度都是64位，应该再长一些会比较安全
- NIST在1997年公开征集新的高级加密标准AES，要求：
 - 安全性能不低于3DES，具有更好的可执行性；必须是分组长度为128位的对称分组密码；必须能够支持128位、192位和256位的密钥
- 在2001年，NIST从15个投选方案中选取了Rijndael作为AES算法
 - Rijndael的作者是比利时的密码学家Joan Daemen和Vincent Rijmen
- AES分组长度128位，密钥长度128位，具有如下特征：对所有已知攻击具有免疫力、设计简单、代码紧凑、执行速度快

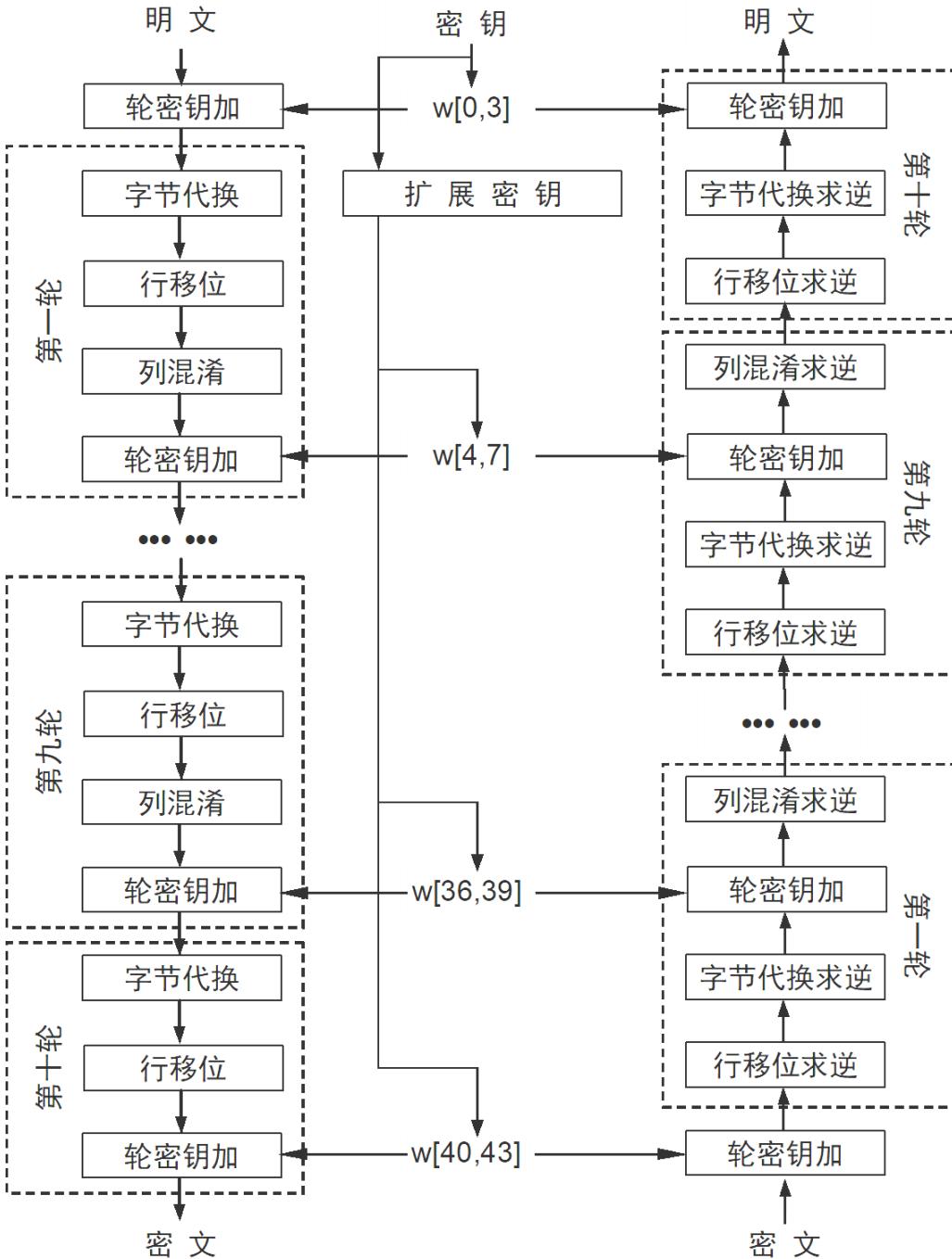
AES的结构

- AES不是Feistel结构
 - AES的每一轮都使用代换和置换并行的处理整个数据分组
- AES结构由四个阶段构成，包括一个置换和三个代换；而且每个阶段均可逆
 - 字节代换：用一个S盒完成分组中的字节代换
 - 行移位：一个简单的置换
 - 列混淆：算术代换
 - 轮密钥加：利用当前分组和扩展密钥的一部分进行按位异或

AES 的加密和解密

● 算法结构简单

- 无论加解密，算法从轮密钥加开始，接着执行九轮迭代，每轮包含四个阶段，最后执行包含三个阶段的最后一轮
- 密钥被扩展成44个32位字所构成的数组w[i]
 - 每轮加密解密过程中，有四个32位字的密钥作为该轮的轮密钥
- 仅在轮密钥加阶段使用密钥
 - 轮密钥加实际上是一种Vernam密码；轮密钥加和混淆等交替使用，是非常有效和安全的

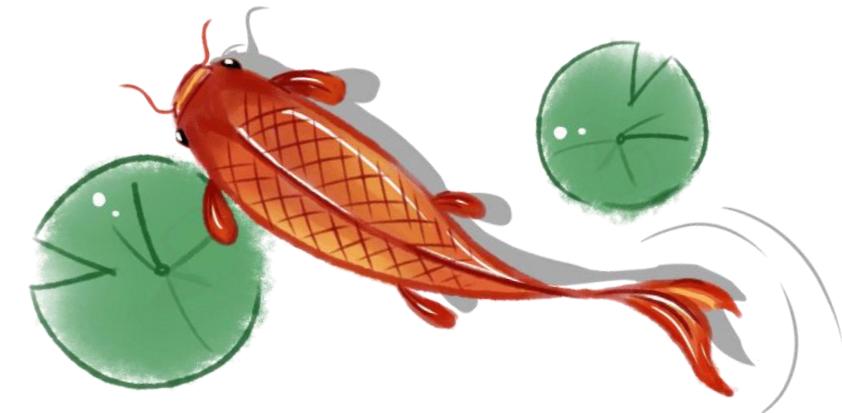


其他对称密码算法

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



非对称密码[1]: 算法原理



非对称密码算法

- 对称密钥密码系统的缺陷
 - 密钥必须经过安全的信道分配
 - 无法用于数字签名
 - 密钥管理复杂，密钥的数量： $O(n^2)$
- 1976年，Whitfield Diffie和Martin Hellman提出了非对称密钥密码（一个公钥、一个私钥），也称公钥密码
- 公钥密码和之前四千多年来所使用的所有密码学方法都不同，是密码学中的一个惊人的成就，是密码学历史上唯一的一次真正的革命
- 公钥密码是基于数学函数而不是代换和置换

公钥密码体制

- 公钥密码体制有6个组成部分。

- 明文：可读的信息，做为加密算法的输入。
- 加密算法：对明文进行的各种变换。
- 公钥/私钥：一个用于加密，一个用于解密。
加密算法执行的变换依赖于公钥和私钥。
- 密文：加密算法的输出，不可读信息。密文依赖于明文和密钥，不同的密钥产生不同的密文。
- 解密算法：根据密文和相应的密钥，产生出明文。

符号说明

- 会话密钥 K_s
- 用户A的公钥 K_{Ua}
- 用户A的私钥 K_{Ra}
- $E_{K_{Ua}}[P]$: 用A的公钥对明文P加密
- $E_{K_{Ra}}[P]$: 用A的私钥对明文P加密
- $D_{K_{Ua}}[C]$: 用A的公钥对密文C解密
- $D_{K_{Ra}}[C]$: 用A的私钥对密文C解密

公钥密码体制

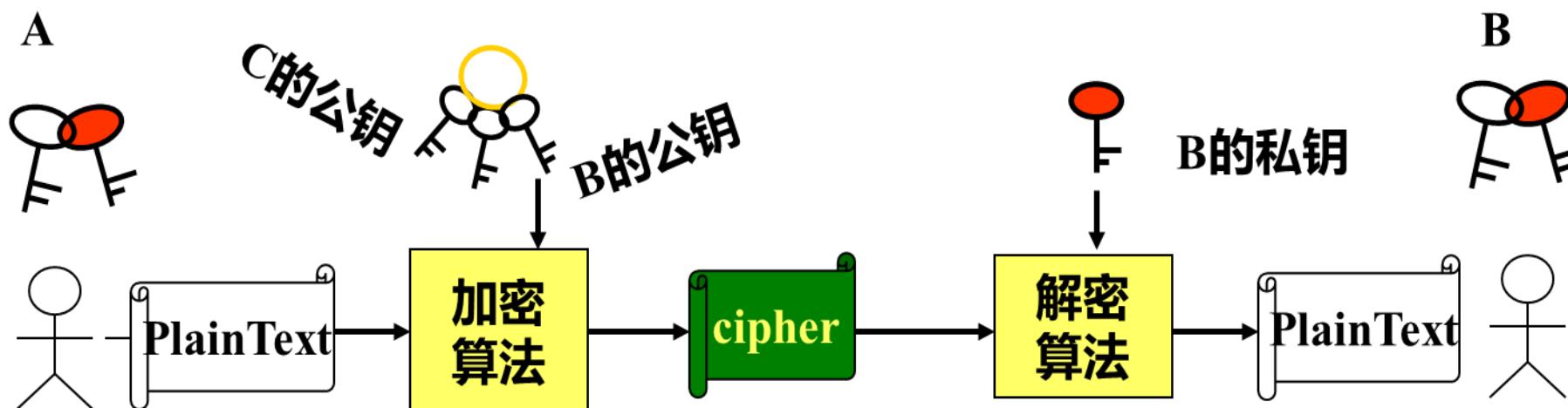
- 每个用户产生一对密钥：用于加密和解密。
其中一个密钥存于公开的寄存器或者文件中，即公钥；另外一个密钥是私有的，称为私钥
 - 公钥公开，用于加密和验证签名
 - 私钥保密，用作解密和签名
- 例如：
 - Bob给Alice发消息，用Alice的公钥对信息加密
 - Alice收到消息后，用自己的私钥解密
 - 由于只有Alice拥有自己的私钥，所以其他人都不能解密出其信息来

公钥密码体制

- 利用这种方法，通信各方皆可以访问公钥，而私钥是个通信方在本地产生的，所以不必进行分配
- 只要系统控制了私钥，那么它的通信是安全的
- 任何时刻，系统都可以改变自己的私钥，而公布相应的公钥代替原来的公钥

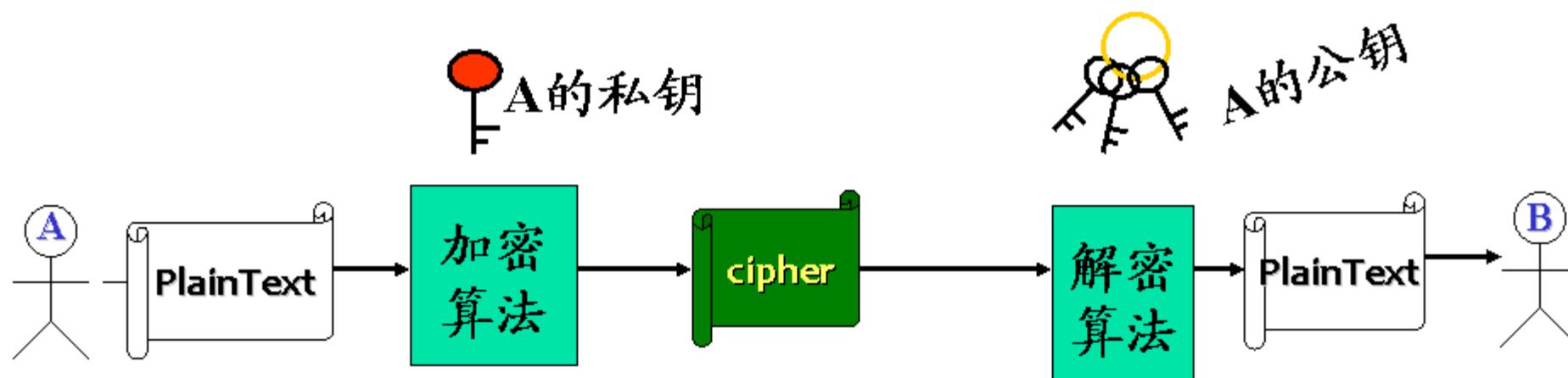
公钥密码系统的加密原理

- 每个通信实体有一对密钥（公钥,私钥）。公钥公开，用于加密和验证签名；私钥保密，用作解密和签名
 - A向B发送消息，用B的公钥加密
 - B收到密文后，用自己的私钥解密
 - 任何人向B发信息都使用同一个密钥(B的公钥)加密。没有人可以得到B的私钥，只有B可以解密



公钥密码系统的签名原理

- A向B发送消息，用A的私钥加密(签名)
- B收到密文后，用A的公钥解密(验证)



公钥密码算法的表示

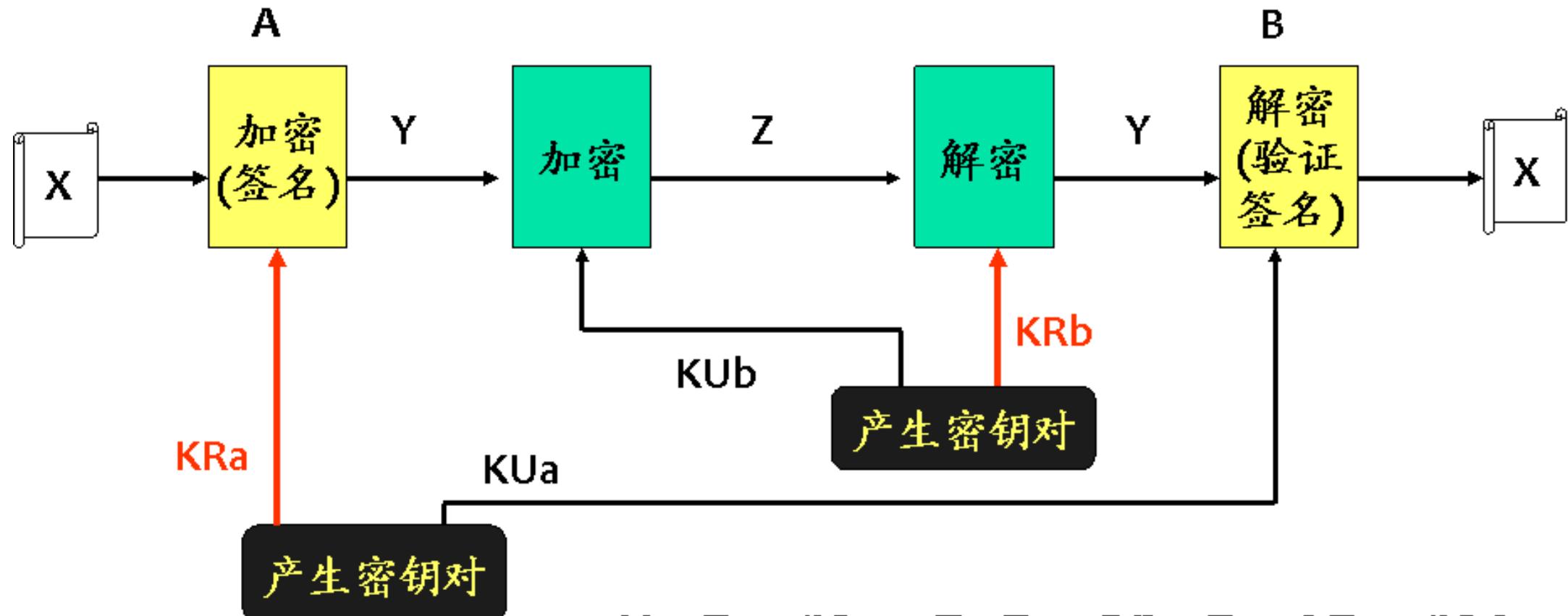
- 对称密钥算法

- 密钥：会话密钥(K_s)
- 加密函数： $C = E_{K_s}[P]$
- 对密文C，解密函数： $D_{K_s}[C]$,

- 公开密钥算法

- A(K_{Ua} , K_{Ra})向B (K_{Ub} , K_{Rb})发送信息：
- 加密： $C = E_{K_{Ub}}[P]$ (用B的公开密钥加密)
- 解密： $P = D_{K_{Rb}}[C]$
- 签名： $E_{K_{Ra}}[P]$ (用A的私有密钥加密)
- 验证： $D_{K_{Ua}}[C]$

数字签名和加密同时使用



$$Y = E_{KRa}(X), \quad Z = E_{KUb}[Y] = E_{KUb}[E_{KRa}(X)]$$

$$Y = D_{KRb}(Z), \quad X = D_{KUa}[Y] = D_{KUa}[D_{KRb}(Z)]$$

公钥密码的数学原理：陷门单向函数

- 公钥密码系统是基于陷门单向函数的概念
- 单向函数是求逆困难的函数；单向陷门函数，是在不知陷门信息下求逆困难的函数，当知道陷门信息后，求逆是易于实现的
 - 单向陷门函数 $f(x)$ ，必须满足以下三个条件：
 - ①给定 x ，计算 $y=f(x)$ 是容易的
 - ②给定 y ，计算 x 使 $y=f(x)$ 是困难的
(所谓计算 $x=f^{-1}(y)$ 困难是指计算上相当复杂已无实际意义)
 - ③存在 δ ，已知 δ 时对给定的任何 y ，若相应 x 存在，则计算 x 使 $y=f(x)$ 是容易的
- 仅满足① ②两条为单向函数；第③条为陷门性， δ 称为陷门信息

公钥密码系统的应用

- 公钥密码系统有三种用途：

- 加密/解密
- 数字签名：如果电子文件都需要签名，如何能够确保数字签名是出自某个特定人，而且通信双方无异议
 - 发送方用自己的**私钥**签署报文，接收方用对方的公钥验证对方的签名
- 密钥交换：双方协商会话密钥，用于对称密钥数据加密

公钥密码算法	加密/解密	数字签名	密钥交换
RSA	Y	Y	Y
Diffie-Hellman	N	N	Y
DSA	N	Y	N

公钥密码分析

- 与对称密码相同，公钥密码也容易受到穷举攻击
 - 攻击者可以利用公钥对所以可能的密钥加密，并与所传送的密文对比，从而可以解密任何消息
 - 公钥体制使用的是某种可逆的数学函数，函数值的复杂性可能不是密钥长度的线性增长，而是指数或者更快的增长速度
- 对应穷举攻击，解决方法也是使用长密钥
 - 为了抗击穷举攻击，密钥必须足够长，为了便于实现，密钥又要足够短
- 在实际应用中，公钥密码目前仅局限于密钥管理和数字签名

对称密码 vs 非对称密码

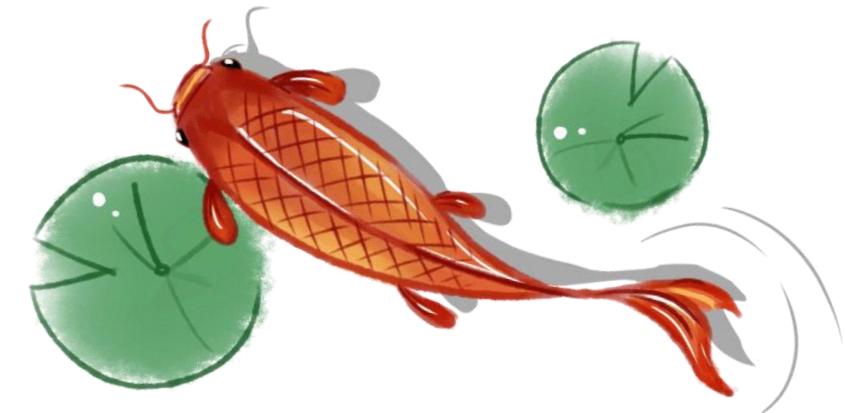
- 非对称的公钥密码学，使用两个独立的密钥，这对于密钥分配、数字签名、认证技术等有深远影响；但存在一些误解
- 误解一：从密码分析角度看，公钥密码比传统密码更安全
 - 任何加密方法的安全性依赖于密钥长度和破译密文所需要的计算量
 - 从抗击密码分析的角度看，不能简单地说传统密码和公钥密码那个更安全
- 误解二：公钥密码是一种通用方法，传统密码已经过时
 - 由于公钥密码需要大量计算，仅限于密钥管理和签名这类应用中，所以基本不太可能取代传统密码
- 误解三：传统密码中与密钥分配中心的握手是一件异常麻烦的事情，而公钥密码实现密钥分配则是非常简单的
 - 使用公钥密码也需要某种形式的协议，通常也包括一个中心代理，所包含的处理过程即不比传统密码简单，也不比其更有效

对称密码和公钥密码

	对称密码	公钥密码
一般要求	加密和解密使用相同的密钥	加密和解密使用相同的密码算法，但使用不同的密钥
	收发双方必须共享密钥	发送方拥有加密或解密的密钥，而接收方只拥有另一个密钥。
安全要求	密钥必须是保密的	两个密钥之一必须是保密的
	若没有其它信息，则解密消息是不可能或者至少是不可行的	若没有其他信息，则解密消息是不可能或者至少是不可行的。
	知晓算法和若干密文不足以确定密钥	知道算法和其中一个密钥以及若干密文不足以确定另外一个密钥



非对称密码[2]: 数论基础



数论基础：素数

- 素数：整数 $P > 1$ 是素数，当且仅当它只有因子+1和+P。
 - 100以内的素数
 - 2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89
97
- 设 P 是所有素数的集合，则任意正整数可以唯一地表示为：
- a与b的最大公因数：gcd (a, b)
 - gcd(20, 24)=4 , gcd (15, 16)=1
- 如果 $\text{gcd}(a, b)=1$ ，称a与b 互素

数论基础：模运算

● 模运算 mod

- $a = q n + r$ 其中 $0 \leq r < n$; $q = [a/n]$;
则有 $r = a \bmod n$
 - $[x]$ 表示小于或等于 x 的最大整数
- 如果 $(a \bmod n) = (b \bmod n)$,
则称 a 与 b 模 n 同余,
记为 $a \equiv b \pmod{n}$
 - 例如, $2^3 \equiv 8 \pmod{5}$, $8 \equiv 1 \pmod{7}$

数论基础：模运算

- 模运算对加法和乘法是可交换、可结合、可分配的

- $(a+b) \bmod n = ((a \bmod n) + (b \bmod n)) \bmod n$
- $(a-b) \bmod n = ((a \bmod n) - (b \bmod n)) \bmod n$
- $(a \times b) \bmod n = ((a \bmod n) \times (b \bmod n)) \bmod n$
- $(a \times (b+c)) \bmod n = ((a \times b) \bmod n + (a \times c) \bmod n) \bmod n$

- 幂, 模运算 $ma \bmod n$

- $m^2 \bmod n = (m \times m) \bmod n = (m \bmod n)^2 \bmod n$
- $m^4 \bmod n = (m^2 \bmod n)^2 \bmod n$
- $m^8 \bmod n = ((m^2 \bmod n)^2 \bmod n)^2 \bmod n$
- $m^{25} \bmod n = (m \times m^8 \times m^{16}) \bmod n$

数论基础：欧拉函数

- 欧拉函数 $\phi(n)$ ：
 n 是正整数, $\phi(n)$ 是比 n 小且与 n 互素的正整数个数
 - $\phi(3)=|\{1, 2\}| = 2$
 - $\phi(4)=|\{1, 3\}| = 2$
 - $\phi(5)=|\{1, 2, 3, 4\}| = 4$
 - $\phi(6)=|\{1, 5\}| = 2$
 - $\phi(7)=|\{1, 2, 3, 4, 5, 6\}| = 6$
 - $\phi(10)=|\{1, 3, 7, 9\}| = 4$
- 如果 p 是素数，则 $\phi(p)=(p-1)$
- 如果 p, q 是素数，则 $\phi(pq)=\phi(p)\phi(q) = (p-1)(q-1)$

数论基础：欧拉定理

● 欧拉定理

- 若整数 m 和 n 互素，则 $m^{\phi(n)} \equiv 1 \pmod{n}$
等价形式 $m^{\phi(n)+1} \equiv m \pmod{n}$

● 例如：

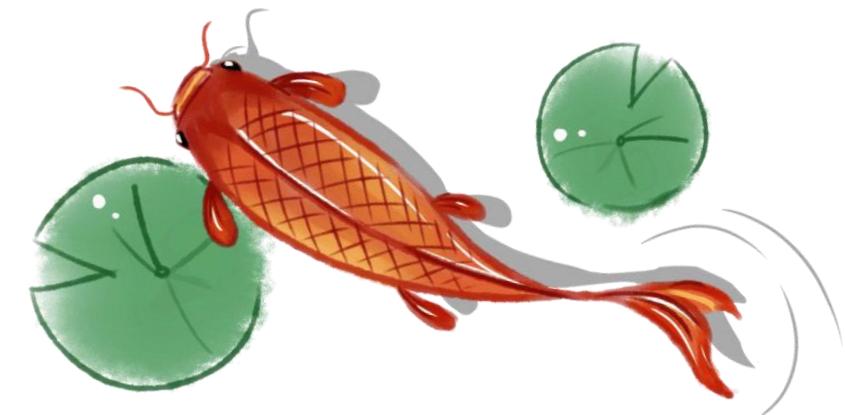
- $m=3, n=10; \phi(10)=4; m^{\phi(n)}=3^4=81; 81 \pmod{10}=1$
- 即： $81 \equiv 1 \pmod{10}; 3^{4+1}=243 \equiv 3 \pmod{10}$

● 推论：

- 给定两个素数 p, q , $p <> q$, 两个整数 n, m , 使得 $n=pq, 0 < m < n$; 则对于任意整数 k , 下列关系成立: $m^{k\phi(n)+1} \equiv m \pmod{n}$

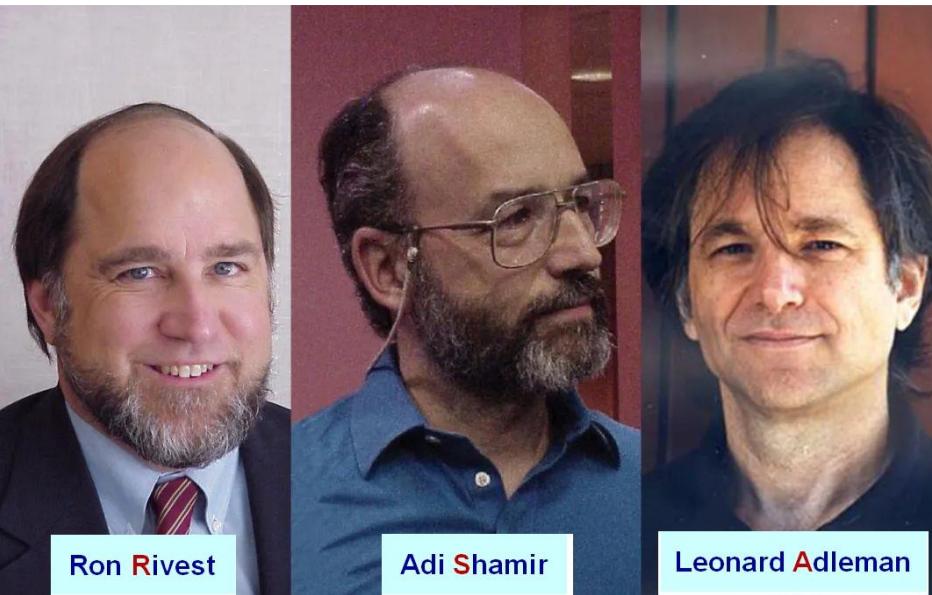


非对称密码[3]: RSA算法



RSA算法简介

- MIT的Ron **Rivest**, Adi **Shamir** , Leonard **Adleman**于1977年提出，于1978年首次发表的RSA算法，是最早满足要求的、被广泛接收、被实现的通用公钥密码算法之一
- RSA体制是一种分组密码，其明文和密文都是0~n-1之间的整数，通常n的大小为1024位二进制数或者309位十进制数



RSA算法：密钥的产生

● 密钥产生

- 取两个大素数 p, q , 保密;
 - 计算 $n=pq$, 公开 n ;
 - 计算欧拉函数 $\phi(n) = (p-1)(q-1)$;
 - 任意取一个与 $\phi(n)$ 互素的小整数 e ,
即 $\gcd(e, \phi(n))=1$; $1 < e < \phi(n)$
 - 寻找 d , $d < \phi(n)$, 使得
 $de \equiv 1 \pmod{\phi(n)}$, 即 $de = k\phi(n) + 1$
 - 公开 (e, n)
 - 将 d 保密, 丢弃 p, q 。
- 公开密钥: $KU=\{e, n\}$
 - 秘密密钥: $KR=\{d, n\}$

- ④ 设: $p=7, q=17$
- ④ 则: $n=119$
- ④ $\Phi(n)=6 \times 16=96$
- ④ 选择 $e=5$
- ④ $5d=k \times 96+1$
- ④ 令 $k=4$, 得到
 $d=77$
- ④ 故知道:
 - ⑤ $KU = \{5, 119\}$
 - ⑤ $KR = \{77, 119\}$

RSA 算法：加密/解密

- 利用：公钥 $KU=\{\underline{e}, n\}$ 和私钥 $KR=\{\underline{d}, n\}$
- 加密过程
 - 把待加密的内容分成k比特的分组， $k \leq \log_2 n$ ，并写成数字，设为M，则： $\underline{C= M^e \bmod n}$
 - 例如： $C=M^5 \bmod 119$
- 解密过程
 - $\underline{M = C^d \bmod n}$
 - 例如： $M=C^{77} \bmod 119$

RSA 算法的证明

- 试证明：解密过程是正确的

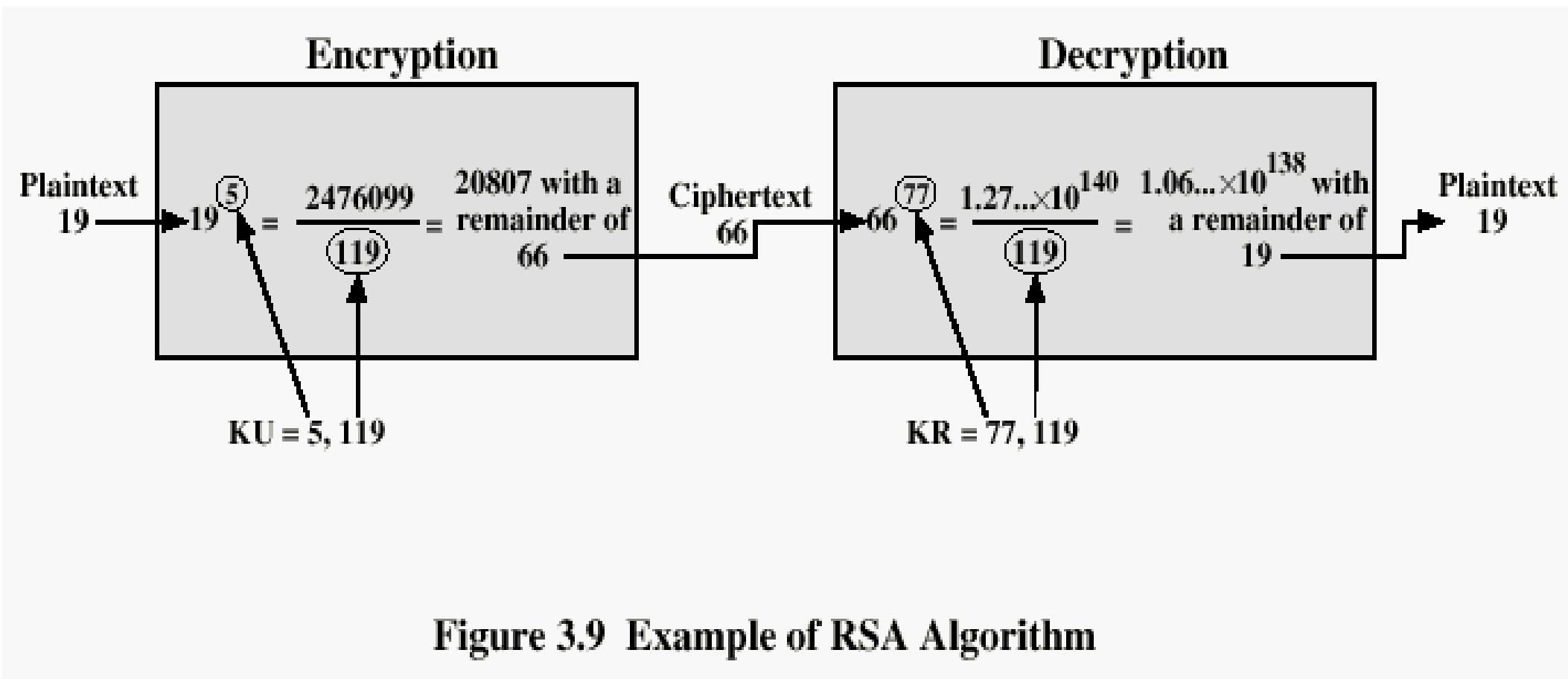
- 证明：

$$\begin{aligned} M &= C^d \bmod n \\ &= (M^e \bmod n)^d \bmod n \\ &= M^{ed} \bmod n \end{aligned}$$

即 $M^{ed} \equiv M \bmod n$

- 根据欧拉定理推论: $M^{k\phi(n)+1} \equiv M \bmod n$,
得到 $ed = k\phi(n)+1$

RSA加密过程举例



- 用RSA算法对下面数据实现加密和解密。
 - $p=5$; $q=11$; $e=3$; $M=9$
- 密钥产生
 - 取两个大素数 p, q , 保密;
 - 计算 $n=pq$, 公开 n ;
 - 计算欧拉函数
 $\phi(n) = (p-1)(q-1)$;
 - 任意取一个与 $\phi(n)$ 互素的小整数 e ,
 即 $\gcd(e, \phi(n)) = 1; 1 < e < \phi(n)$
 - 寻找 d , $d < \phi(n)$, 使得
 $de \equiv 1 \pmod{\phi(n)}$,
 即 $de = k\phi(n) + 1$
 - 公开 (e, n)
 - 将 d 保密, 丢弃 p, q 。
- 公开密钥: $KU=\{e, n\}$
- 秘密密钥: $KR=\{d, n\}$

RSA算法举例1

- 设: $p=5, q=11$
- 则: $n=5*11=55$
- $\Phi(n)=(5-1)*(11-1)=4*10=40$
- 因为 $e=3$
 根据 $de = k\phi(n) + 1$
 故得 $3d=k \times 40 + 1$
- 令 $k=2$, 得到 $d=27$
- 故知道:
 - $KU = \{3, 55\}$
 - $KR = \{27, 55\}$

- 加密过程
 $C=M^e \pmod{n}$
 $=9^3 \pmod{55} = 729 \pmod{55} = 14$
- 解密过程
 $M=C^d \pmod{n}$
 $=14^{27} \pmod{55} = 9$

RSA算法举例2

- 假设Alice需要将明文“key”通过RSA加密后传递给Bob。

- 第一步：设计公钥和密钥

- 令 $p=3$, $q=11$, 得出 $n=p \times q=3 \times 11=33$
- $\Phi(n)=(p-1)(q-1)=2 \times 10=20$
- 取 $e=3$, 则 $e \times d \equiv 1 \pmod{\Phi(n)}$, 即 $3 \times d \equiv 1 \pmod{20}$; 取 $d=7$
- 从而可以设计出一对公私密钥,
 - 加密密钥（公钥）为： $KU=(e,n)=(3,33)$
 - 解密密钥（私钥）为： $KR=(d,n)=(7,33)$

RSA算法举例2

● 第二步：明文信息数字化

- 假定明文英文字母编码表为按字母顺序排列数值
- 则得到分组后的key的明文信息为： 11， 05， 25。

字母	A	B	C	D	E	F	G	H	I	J	K	L	M
码值	01	02	03	04	05	06	07	08	09	10	11	12	13
字母	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
码值	14	15	16	17	18	19	20	21	22	23	24	25	26

RSA算法举例2

● 第三步：明文加密

- Alice用加密密钥(3,33) 将数字化明文分组信息加密成密文
- 由 $C \equiv M^e \pmod{n}$ 得到相应的密文信息为：
11, 26, 16

$$C_1 \equiv (M_1)^e \pmod{n} = 11^3 \pmod{33} = 11$$

$$C_2 \equiv (M_2)^e \pmod{n} = 5^3 \pmod{33} = 26$$

$$C_3 \equiv (M_3)^e \pmod{n} = 25^3 \pmod{33} = 16$$

● 第四步：密文解密

- Bob收到密文后，若将其解密，只需要计算 $m_i = C_i^d \pmod{n}$
 - $M_1 = 11^7 \pmod{33} = 11$; $M_2 = 26^7 \pmod{33} = 05$; $M_3 = 16^7 \pmod{33} = 25$
- 得到明文信息为：11, 05, 25; 根据编码表将其转换为“key”

RSA 算法的安全性

- 对RSA算法的攻击方法：蛮力攻击、数学攻击、计时攻击
- 蛮力攻击：对所有密钥都进行尝试
- 数学攻击：有多种数学攻击方法，他们的实质是两个素数乘积(n)的因子分解
- 计时攻击：类似于通过观察他人转动保险柜拨号盘的时间长短来猜测密码；攻击者可以通过记录计算机解密消息所用的时间来确定私钥
 - 计时攻击不仅可以用于攻击RSA，还可以用于攻击其它公钥密码系统，由于这种攻击的完全不可预知性以及它仅仅依赖明文，所以计时攻击具有很大的威胁

RSA 算法的安全性

- 大数因子分解是数论中的一个难题，因子分解的进展可用MIPS年描述计算的代价
 - MIPS年是指一台每秒执行百万条指令的处理器运行一年

十进制数位数	近似比特数	完成日期	MIPS年
100	332	1991年4月	7
110	365	1992年4月	75
120	398	1993年6月	830
129	428	1994年4月	5000
130	431	1996年4月	1000
140	465	1999年2月	2000
155	512	1999年8月	8000

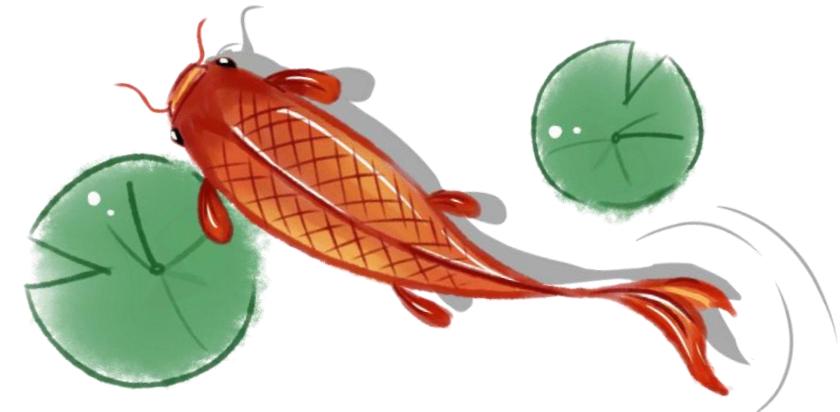
RSA 算法的性能

- 与其它密码体制一样，RSA抗穷举攻击的方法也是使用大密钥空间，但是密钥越大，系统运行速度越慢
 - 软件实现比DES慢100倍
 - 硬件实现比DES慢1000倍

	512位	768位	1024位
加密	0.03	0.05	0.08
解密	0.16	0.48	0.93
签名	0.16	0.52	0.97
验证	0.02	0.07	0.08



非对称密码[4]: DH密钥交换算法



Diffie-Hellman密钥交换算法

- 1976年，Diffie-Hellman第一个发表的公开密钥算法，被称为Diffie-Hellman密钥交换算法
- Diffie-Hellman密钥交换算法的目的是使两个用户能够安全地交换密钥，该算法本身也只局限于进行密钥交换
- Diffie-Hellman密钥交换算法的有效性在于计算离散对数非常困难

离散对数

- 对数是指数的反函数: $b = a^i \Rightarrow i = \log_a b$
- 本原根 (Primitive Root)
 - a 是素数 p 的一个本原根, 如果
 - $a \bmod p, a^2 \bmod p, \dots, a^{p-1} \bmod p$ 是1到 $p-1$ 的排列, 即各不相同, 是整数1到 $p-1$ 的一个置换。

$p=19, a^i \bmod p, i=1,2,3,\dots,18$																		
a	a^2	a^3	a^4	a^5	a^6	a^7	a^8	a^9	a^{10}	a^{11}	a^{12}	a^{13}	a^{14}	a^{15}	a^{16}	a^{17}	a^{18}	
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
2	4	8	16	13	7	14	9	18	17	15	11	3	6	12	5	10	1	
3	9	8	5	15	7	2	6	18	16	10	11	14	4	12	17	13	1	
4	16	7	9	17	11	6	5	1	4	16	7	9	17	11	6	5	1	
5	6	11	17	9	7	16	4	1	5	6	11	17	9	7	16	4	1	

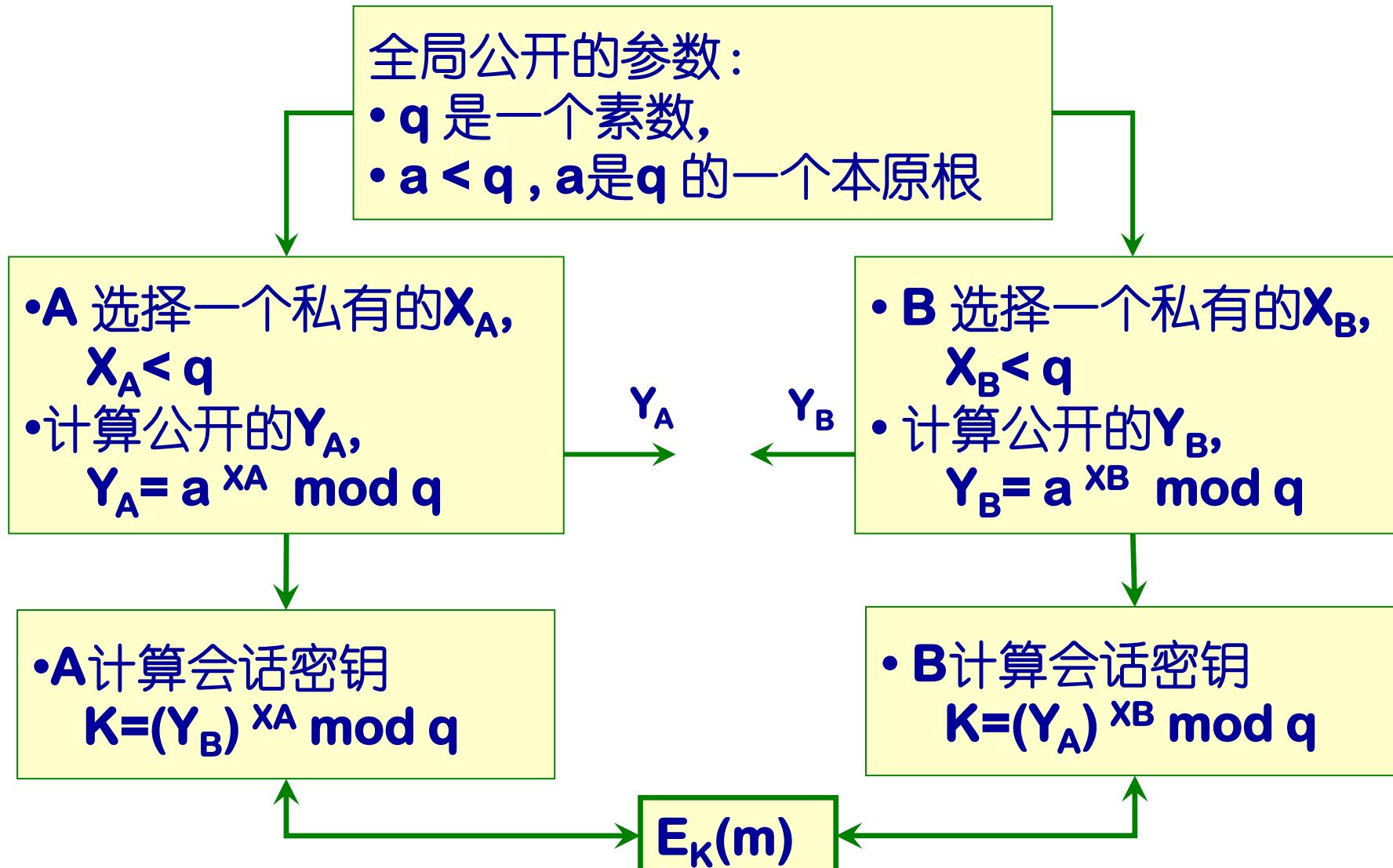
离散对数

- 对于整数 b ($b < p$) 和素数 p 的一个本原根 a ,
可以找到一个唯一的指数 i , 使得:
 $b \equiv a^i \pmod{p}$, 其中 $0 \leq i \leq (p-1)$

i 称为 b 的以 a 为底 模 p 的离散对数或指数,
记为 $\text{ind}_{a,p}(b)$

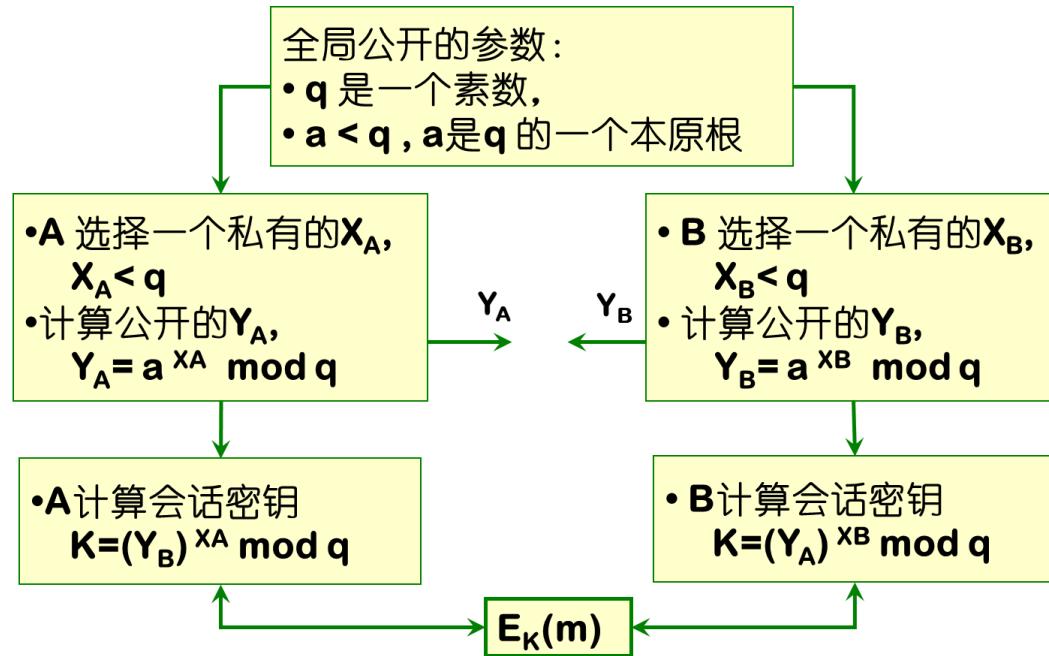
- $\text{ind}_{a,p}(1)=0$, 因为 $a^0 \pmod{p} = 1 \pmod{p} = 1$;
 - $\text{ind}_{a,p}(a)=1$, 因为 $a^1 \pmod{p} = a \pmod{p} = a$;
-
- 对于 $b = a^x \pmod{p}$
 - 已知 a, x, p , 计算 b 是容易的
 - 已知 a, b, p , 计算 x 是非常困难的

Diffie-Hellman 密钥交换过程



DH算法举例

- 全局公开参数: $q=97$, $a = 5$
(5是97的本原根)
- A选择私钥 $X_A=36$
- B选择私钥 $X_B=58$



- A 计算公钥 $Y_A = 5^{36} \bmod 97 = 50$
- B 计算公钥 $Y_B = 5^{58} \bmod 97 = 44$
- A 与 B 交换公开密钥

- A 计算会话密钥
 - $K = Y_B^{XA} \bmod q = 44^{36} \bmod 97 = 75$
- B 计算会话密钥
 - $K = Y_A^{XB} \bmod q = 50^{58} \bmod 97 = 75$

其他公钥密码算法

- DSA

- 1991年,NIST 提出了数字签名算法(DSA),并用于数字签名标准(DSS)
- DSA只能用于数字签名, 算法的安全性是基于计算离散对数的难度
- 但是DSA招致大量的反对:
 - DSA 不能用于加密或密钥分配
 - DSA是由 NIST研制的, 可能有后门
 - DSA的选择过程不公开, 提供的分析时间不充分
 - DSA比RSA慢(10—40倍)
 - 密钥长度太小(512位)
 - DSA可能侵犯其他专利

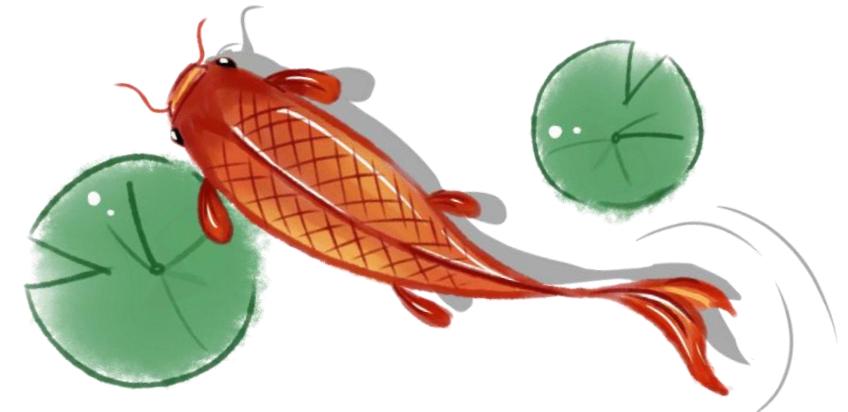
- 椭圆曲线密码系统

- 有限域 $GF(2^n)$
- 运算器容易构造
- 加密速度快
- 更小的密钥长度实现同等的安全性

- RSA是事实上的标准

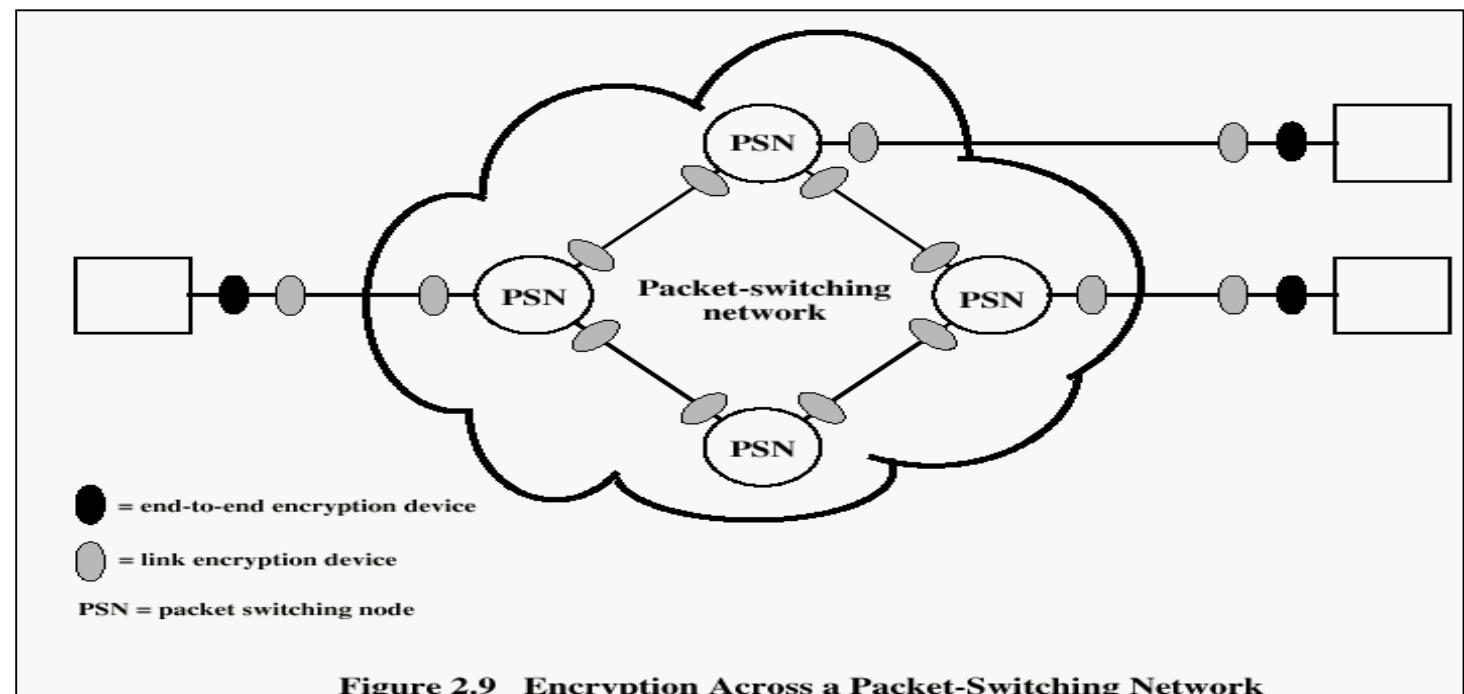


密钥分配[1]: 概述



密码功能的设置

- 对于网络而言，攻击者无处不在
- 加密是所使用的最有力也最常用的方法
- 需要确定的是：加密什么和在什么地方加密
- 存在两种方法：
 - 链路加密
 - 端到端加密

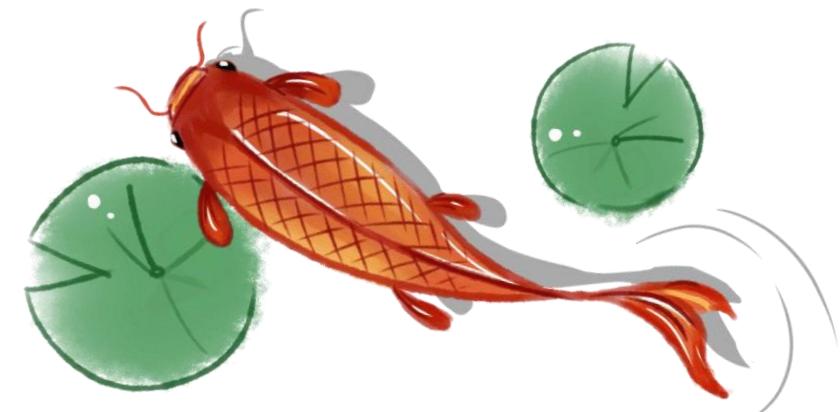


密钥的分配

- 无论是对称密码还是非对称密码，任何密码系统的强度都和密钥分配方法有关
- 密钥分配方法就是将密钥发放给希望交换数据的双方而不让别人知道的方法
 - 情况一：传统的对称密码分配
 - 情况二：非对称密码中的公钥分配
 - 情况三：公钥密码用于传统密码体制的密钥分配



密钥分配[2]: 情况一 传统的对称密码分配



传统的对称密码分配

- 对参与者A和B，传统的对称密钥分配有：
 - 方法一：密钥由A选择，并亲自交给B
 - 方法二：第三方选择密钥后亲自交给A和B
 - 方法三：如果A和B以前或者最近使用过某个密钥，其中一方可用它加密一个新密钥后再发送给另外一方
 - 方法四：A和B与第三方C均有秘密渠道，则C可以将一密钥分别秘密发送给A和B
- 方法一和二需要人工传送密钥，适用于链路加密；对于端到端加密，则使用密钥分配中心

密钥分配中心KDC模式

- 假定每个用户与密钥分配中心KDC共享唯一的一个主密钥
 - 设A要和B建立一个逻辑连接
 - A有一个除了自己只有KDC知道的主密钥Ka
 - B有一个除了自己只有KDC知道的主密钥Kb

密钥分配中心KDC模式

- A和B获得一次性会话密钥Ks的工作过程
 - STEP1: A向KDC请求一个会话密钥以保护与B的逻辑连接；消息中有A和B的标识以及一个临时交互号N1
 - STEP 2: KDC用Ka加密的消息做出响应；此时，A可知：
 - 一次性会话密钥Ks，用于会话
 - 原始请求消息，含N1，使A可以做出反映
 - 此外，还含有两项给B的内容，用Kb加密了一次性会话密钥Ks 和 A的标识符IDa
 - STEP 3: A存下会话密钥Ks备用，将KDC消息中的后两项内容发给B，B也知道会话密钥了
- 这样会话密钥Ks就安全地发给了A和B

面向连接协议的自动密钥分配

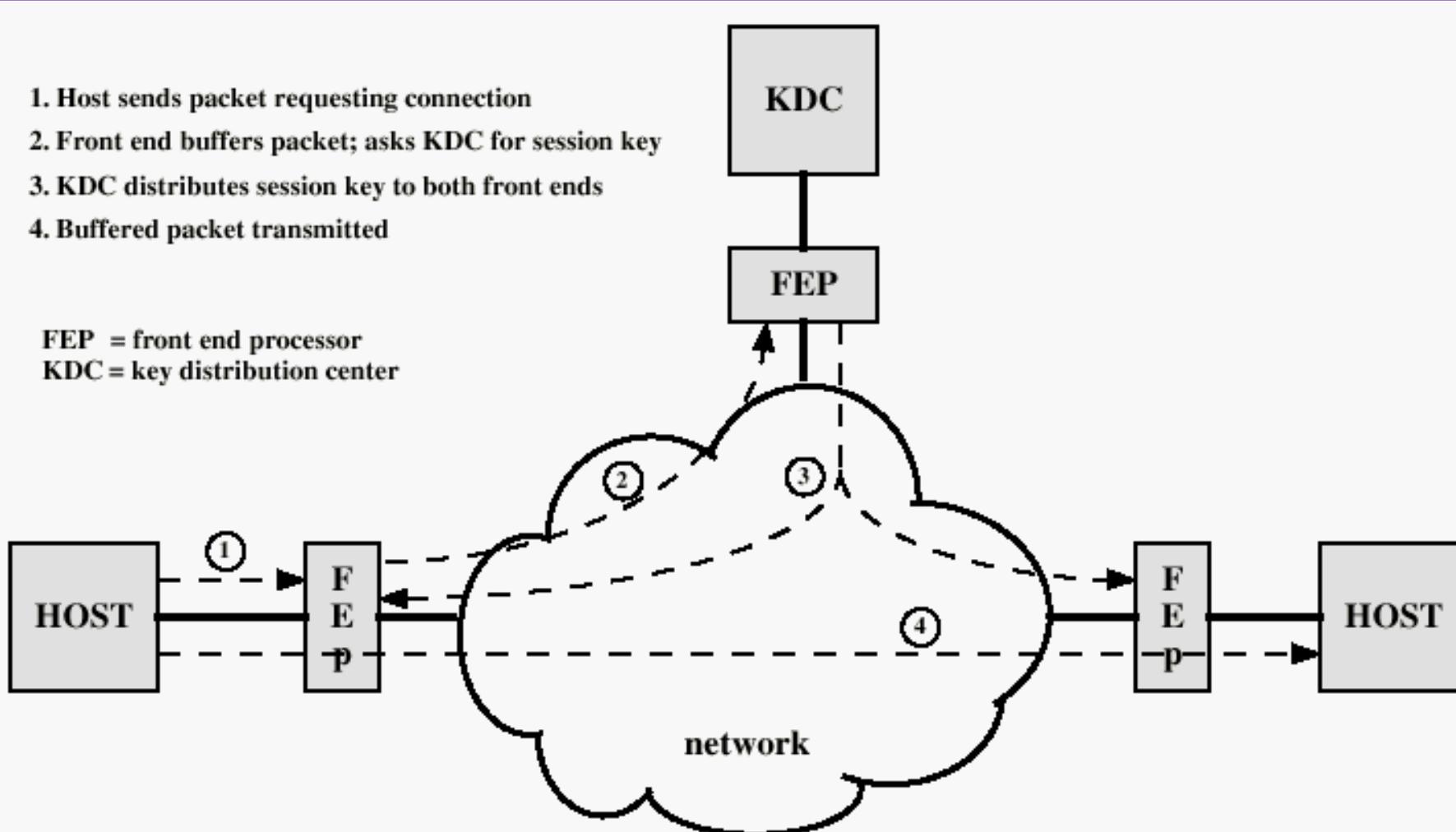


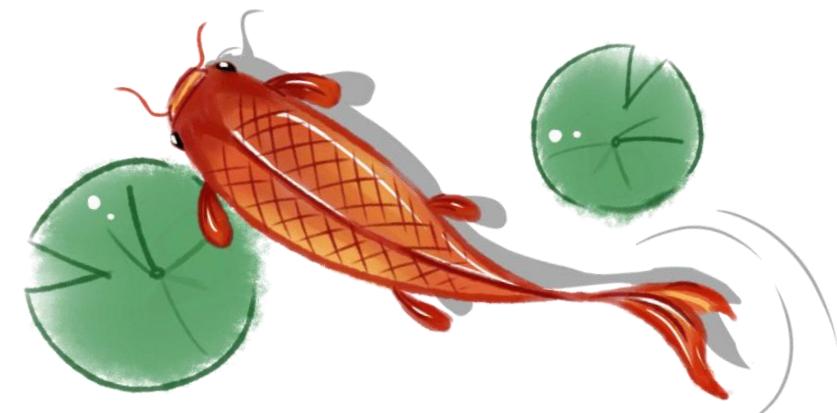
Figure 2.10 Automatic Key Distribution for Connection-Oriented Protocol

密钥的控制

- 在网络规模很大的时候，密钥的分配功能不限定在单个的KDC上面，而是使用层次式的KDC
- 层次式KDC密钥分配使得主密钥分配的代价变小了；如果一个本地KDC出错，或者被攻击了，破坏只是集中在一个区域中，不会影响全局



密钥分配[3]: 情况二 公钥的分配



公钥的分配

- 公钥密码的主要作用之一就是解决密钥分配问题，公钥密码可用于下面两个方面：
 - 公钥的分配
 - 公钥密码用于传统密码体制的密钥分配
- 常用的公钥分配方法有四种：
 - 公开发布
 - 公开可访问目录
 - 公钥授权
 - 公钥证书

方法一：公开发布

- 任一通信方可以将他的公钥发送给另外一个通讯方或者广播给通信各方
 - 例如：目前在电子邮件的认证和保密性方面广泛应用的PGP(pretty good privacy)协议，很多PGP用户就在自己发送的信息之后加入自己的公钥
- 这种方法比较简便，但是它有一个大缺点，就是任何人都可以伪造这种公钥的公开发布

方法二：公开可访问目录

- 维护一个动态可访问的公钥目录可以获得更大程度的安全性；某个可信的实体或组织负责这个公开目录的维护和分配
 - 管理员通过对每个通讯方建立一个目录项<姓名，公钥>来维护目录，管理员定期发布或者更新该目录
 - 每个通讯方通过目录管理员注册一个公钥；
通讯方在任何时候都可以用新的密钥替代当前密钥
 - 通讯方也可以访问电子目录；
当然，它必须拥有从管理员到他的安全认证通道
- 这种方法比公开公钥要安全，但是也存在缺点：一旦攻击者获得或者计算出目录管理员的私钥，带来的危险将很大

方法三：公钥授权

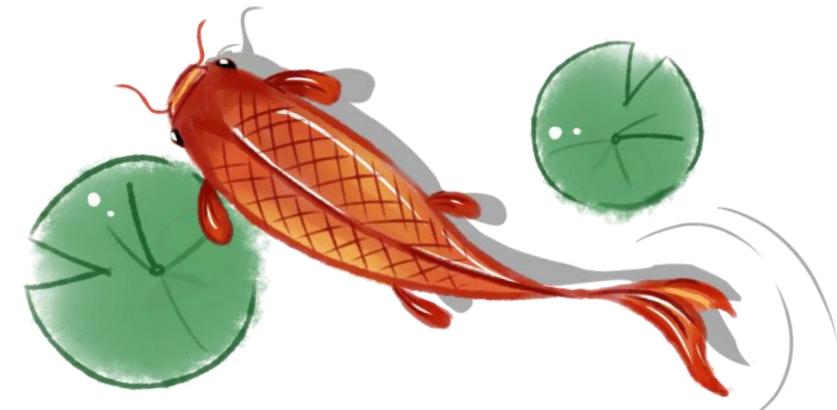
- 公钥授权是一种更严格的方法，它的工作过程：
 - A发送一条带有时间戳的消息给公钥管理员，请求B的当前公钥
 - 管理员给A发送一条用其私钥KR加密的消息，A可以用管理员的公钥解密。这条消息中包含：B的公钥、原始请求、原始时间戳
 - A保存B公钥，并将A的表示和临时交互号N1发给B
 - 与A一样，B使用同样的方法从管理员那里得到A的公钥。
 - A与B已经可以通信了，再通过核对临时交互号确认各自的身份后，安全的通信机制就建立了
- 公钥授权的缺点在于公钥管理员就会成为系统的瓶颈
 - 只要用户之间通信，就必须向目录管理员申请对方的公钥

方法四：公钥证书

- Kohnfelder最早提出不通过管理员，而用整数来交换密钥；此方法与公钥授权的安全性相同
 - 证书包含者公钥和其它一些信息，由证书管理员产生，并发给拥有相应私钥的通讯方
 - 通信一方通过传递证书将密钥信息传递给另外一方，其它通信各方可以验证该证书确实是由于证书管理员发出的
- 这种方法需要满足下面的要求：
 - 任何通信方可以读取证书并确定证书拥有者的姓名和公钥
 - 任何通信方可以验证该证书出自证书管理员，而不是伪造的
 - 只有证书管理员才可以产生并更新证书
 - 任何通信方可以验证证书的当前性



密钥分配[4]: 情况三 利用公钥分配传统密码的密钥



利用公钥分配传统密码的密钥

- 由于公钥密码速度较慢，几乎没有用户愿意在通信中完全使用公钥密码，因此公钥密码更适合作为传统密码中实现密钥分配的一种手段。
- 下面介绍三种密钥分配方法
 - 简单的密钥分配方法
 - 具有保密性和真实性的密钥分配方法
 - 混合方法

简单的密钥分配

- Merkle提出了一种极其简单的方法
- 例如：A要和B通信，则执行：
 - A产生公/私钥对{K_{Ua}, K_{Ra}}，并将含有K_{Ua}和A表示的消息发给B。
 - B产生秘密钥K_s，用A的公钥对K_s加密后传给A。
 - A计算DK_{Ra}[E_{K_{Ua}}[K_s]]得到秘密钥K_s。
 - 这样A和B就可以利用K_s和对称密码进行安全通信。
- 不过，这种协议容易受到主动攻击

具有保密性和真实性的密钥分配

- 下面这种方法即可以抗击主动攻击也可以抗击被动攻击
 - 假设A和B已经交换了公钥
 - A用B的公钥对含有A标识和临时交互号N1的消息加密，并发给B
 - B发送一条用A的公钥加密的消息，包括A的N1和B的临时交互号N2
 - 只有B能够解密A发来的消息，所以本条消息中的N1可以使A确信其通信伙伴是B
 - A用B的公钥对N2加密，并返回B，这样可使B确信其通信伙伴是A
 - A选择密钥Ks，并将 $M = E_{KUb}[E_{KRa}[Ks]]$ 发送给B
 - B计算 $D_{KUa}[D_{KRb}[M]]$ 得到密钥Ks
 - A和B可以利用对称密码进行安全通信了
- 利用公钥密码进行密钥分配，也需要密钥分配中心KDC，KDC与每个用户共享一个主密钥，通过该主密钥加密实现会话密钥的分配



Thanks a lot !

Activity is the only road to knowledge!

Computer Network Security @ 2023Fall