回顾上次作业

- ✓ 阅读第4章
- ✓ 思考问题：
  - ✓ 为了保证加密安全性，加密算法本身应该保密。你怎么看？
  - ✓ 密码分析里，什么是统计分析攻击？
  - ✓ 分组密码和序列密码之间的区别是什么？
  - ✓ 对称密码体制和非对称密码体制的不同之处是什么？

第四章 密码学

- ✓ 密码学概述
- ✓ 对称密码体制
- ✓ 非对称密码体制
- ✓ Hash函数与消息认证
- ✓ 数字签名技术
- ✓ 密钥管理技术

4.1 密码学概述

引言

- ✓ 信息安全的主要任务是研究计算机系统和通信网络中信息的保护方法，密码学理论和技术就是其中一个重要的研究领域，可以说密码学是保障信息安全的核心基础。
- ✓ **密码学（cryptology）起源于保密通信技术**，是结合数学、计算机、信息论等学科的一门综合性、交叉性学科。密码学又分为**密码编码学（cryptography）**和**密码分析学（cryptanalysis）**两部分。密码编码学主要研究如何设计编码，使得信息编码后除指定接收者外的其他人都不能读懂。密码分析学主要研究如何攻击密码系统，实现加密消息的破译或消息的伪造。

密码与保密通信

- 这个过程中，箱子和锁的种类、制造的方法就类似于”算法”，开锁的钥匙就是”密钥”。”算法”和传递”密钥”的方式是影响保密程度的关键。

4.1.1密码学发展简史

- 密码学一词源自希腊文“kryptós”（隐藏的）及“gráphein”（书写）两字，即隐密地传递信息。密码的研究和应用已有几千年的历史，其发展经历了以下三个阶段：
  - 古典密码时期
  - 近代密码时期
  - 现代密码时期

古典密码时期

- ✓ 一般认为古典密码时期是从古代到19世纪末
- ✓ 古典密码加解密方法主要基于手工完成，此时期也被称为密码学发展的手工阶段
- ✓ 密文信息一般通过人（信使）来传递
- ✓ 这个时期的经典案例有：公元前2世纪希腊人设计的棋盘密码、公元前约50年古罗马凯撒大帝发明的凯撒密码、美国南北战争时期军队中使用过的栅栏密码等等。目前，这个时期提出的所有密码方法已全部破译。

古典密码

- 棋盘密码
  - 产生于公元前两世纪的希腊。 棋盘密码的密钥是5×5的棋盘，将26个字母放置在里面，其中i和j放在同一个方格中。简单的来说就是把字母排列好，用坐标的形式表现出来。字母是明文，密文便是字母的坐标。
  - 举个例子，HELLO，加密后就是23 15 31 31 34

	1	2	3	4	5
1	a	b	c	d	e
2	f	g	h	i/j	k
3	l	m	n	o	p
4	q	r	s	t	u
5	v	w	x	y	z

古典密码

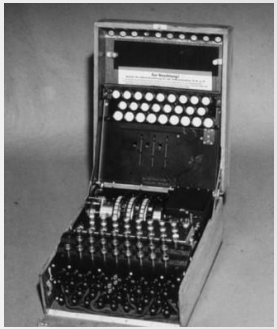
- 栅栏密码
  - 一种置换密码，就是把要加密的明文分成N个一组，然后把每组的第1个字连起来，形成一段无规律的话。一般比较常见的是2栏的栅栏密码。
  - 例如明文：THERE IS A CIPHER

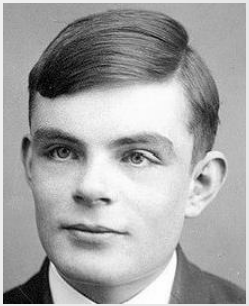
    去掉空格后变为：THEREISACIPHER　　两个一组：TH ER EI SA CI PH ER

    先取出第一个字母：TEESCPE　　再取出第二个字母：HRIAIHR

    连在一起就是：TEESCPEHRIAIHR

近代密码时期



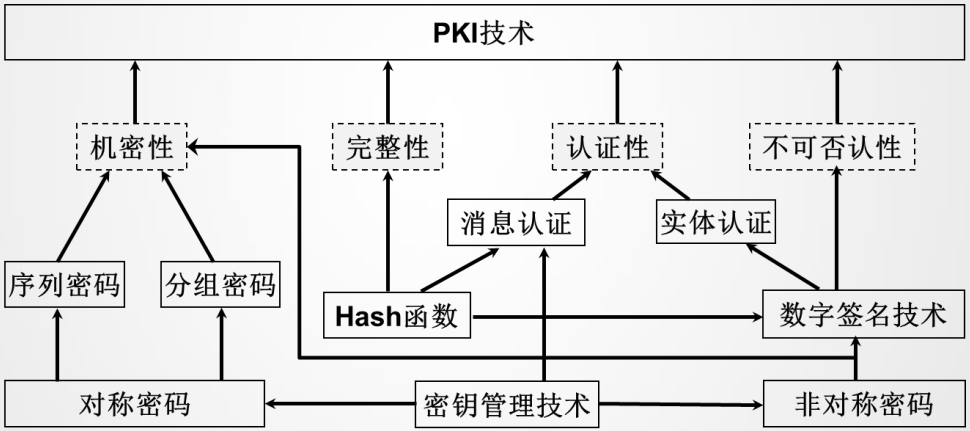


- ✓ 近代密码时期是指20世纪初期到20世纪50年代末
- ✓ 出现了一些利用电动机械设备实现信息加密、解密操作的密码方法
- ✓ 采用电报机发送加密的信息
- ✓ 这个时期的著名密码主要有：美国电话电报公司的Gillbert Vernam设计的Vernam密码、第二次世界大战中使用的Enigma转轮密码机。

现代密码时期

- ✓ 现代密码时期是从20世纪50年代至今
- ✓ 利用计算机技术实现加解密过程。
- ✓ 无线通信、有线通信、计算网络等等方式传递密文
- ✓ 1949年，Shannon的《保密系统的通信理论》标志着密码学作为一门科学的形成。1976年Diffie和Hellman提出公钥密码体制思想，成为密码学发展史上的重要里程碑。这个时期的著名密码主要有：DES（1977年）、RSA（1978年）、AES（2001年）等算法。

现代密码学主要内容


PKI技术
机密性
完整性
认证性
不可否认性
消息认证
实体认证
序列密码
分组密码
Hash函数
数字签名技术
对称密码
密钥管理技术
非对称密码

4.1.2密码体制的基本组成及分类

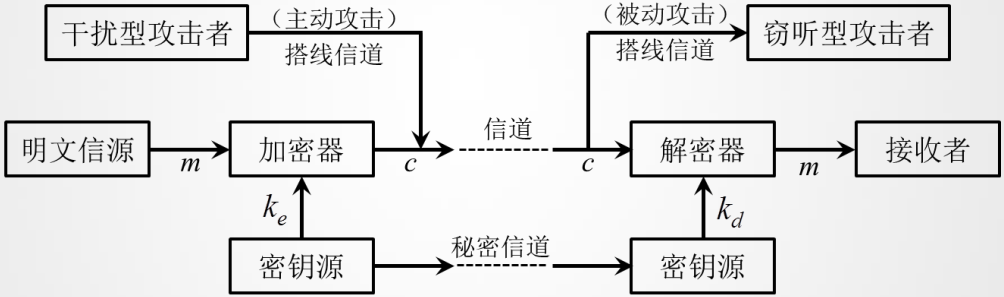


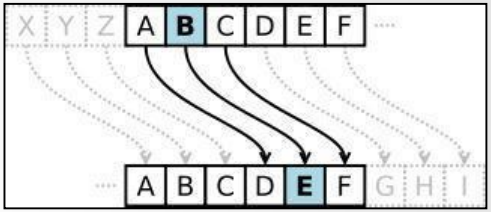
图4-2 保密通信的一般模型

密码体制的组成

一个密码体制（cryptosystem）由五个部分组成：

- ✓ 明文空间M，它是全体明文m 的集合；
- ✓ 密文空间C，它是全体密文c 的集合；
- ✓ 密钥空间K，它是全体密钥k的集合。其中每一个密钥k 均由加密密钥k_e和解密密钥k_d 组成，即$k = (k_e , k_d)$；
- ✓ 加密算法E，是在密钥控制下将明文消息从M 对应到C 的一种变换，即$c = E\ (k_e , m)$；
- ✓ 解密算法D，是在密钥控制下将密文消息从C 对应到M 的一种变换，即$m = D\ (k_d , c)$。

举例



✓ **例4-1 凯撒密码应用示例。**

明文m = It is a secret　　密文c = LWLVDVHFUHW

数学语言可以表示为：$M = C = \{x \mid x \in [0,25] 且 x \in \mathbf{Z}\}$；$k_e = k_d = 3$；

$E(k_e, m) = (m+3) \bmod 26$

$D(k_d, c) = (c-3) \bmod 26$

m	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
+3 mod 26																										
c	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	0	1	2
	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**

举例

- **例4-2 Vernam密码应用示例。**

在对明文加密前，首先将明文编码为（0，1）序列，加密时用明文与密钥进行模2相加，解密时将密文再与密钥模2相加即可。例如：明文为10001 11000，密钥为10010 00101时，加密得到的密文为00011 11101。

在应用Vernam密码时，如果每次使用不同的随机密钥对明文进行加密，则被称为一次一密(one-time pad, OTP)密码。

10001 11000
10010 00101
00011 11101

密码体制的分类

✓ 根据加、解密密钥使用策略不同，可将密码体制分为对称密码体制和非对称密码体制。

对称密码体制（Symmetric Cryptosystem）

如果一个密码体制中的加密密钥k_e和解密密钥k_d相同，或者由其中一个密钥很容易推算出另一个密钥，则称为对称密码体制或单钥密码体制(One-key Cryptosystem)

非对称密码体制(Asymmetric Cryptosystem)

如果在计算上由加密密钥k_e不能推出解密密钥k_d，因此可以将k_e公开，这种密码体制也被称为公钥密码(Public Key Cryptosystem)。

4.1.3密码体制的设计原则

- ✓ **密码学的基本目的就是保障不安全信道上的通信安全。**
- ✓ 密码学领域存在一个很重要的事实：“如果许多聪明人都不能解决的问题，那么它可能不会很快得到解决。”这暗示很多加密算法的安全性并没有在理论上得到严格的证明，只是这种算法思想出来以后，经过许多人许多年的攻击并没有发现其弱点，没有找到攻击它的有效方法，从而认为它是安全的。

衡量密码体制安全性的方法

- **计算安全性（computational security）**。指一种密码系统最有效的攻击算法至少是指数时间的，又称实际保密性（practical secrecy）。
- **可证明安全性（provable security）**。如果密码体制的安全性可以归结为某个数学困难问题，则称其是可证明安全的。
- **无条件安全性（unconditional security）**或者完善保密性（perfect secrecy）。假设存在一个具有无限计算能力的攻击者，如果密码体制无法被这样的攻击者攻破，则称其为无条件安全。

设计原则

一个实用的密码体制的设计应该遵守以下原则：

- ✓ 密码算法安全强度高。就是说攻击者根据截获的密文或某些已知明文密文对，要确定密钥或者任意明文在计算上不可行。
- ✓ 密码体制的安全性不应依赖加密算法的保密性，而应取决于可随时改变的密钥。【柯克霍夫（Kerckhoffs）原则】
- ✓ 密钥空间应足够大。使试图通过穷举密钥空间进行搜索的方式在计算上不可行。
- ✓ 既易于实现又便于使用。主要是指加密函数和解密函数都可以高效地计算。

4.1.4 密码体制的常见攻击形式

- ✓ 穷举攻击：密码分析者通过试遍所有的密钥来进行破译。穷举攻击又称为蛮力攻击，是指攻击者依次尝试所有可能的密钥对所截获的密文进行解密，直至得到正确的明文。
- ✓ 统计分析攻击：密码分析者通过分析密文和明文的统计规律来破译密码。抵抗统计分析攻击的方式是在密文中消除明文的统计特性。
- ✓ 数学分析攻击：密码分析者针对加密算法的数学特征和密码学特征，通过数学求解的方法来设法找到相应的解密变换。为对抗这种攻击，应该选用具有坚实的数学基础和足够复杂的加密算法。

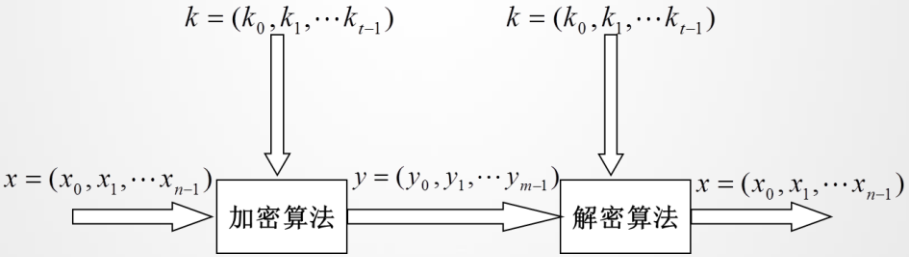
密码分析攻击的类型

- 密码攻击和解密的相似之处在于都是设法将密文还原成明文的过程，根据密码分析者可获取的信息量不同，常见的密码分析攻击包括以下4种类型：
  - 唯密文攻击（ciphertext only attack）
  - 已知明文攻击（known plaintext attack）
  - 选择明文攻击（chosen plaintext attack）
  - 选择密文攻击（chosen ciphertext attack）

4.2 对称密码体制

4.2.1 分组密码

- **分组密码**（Block Cipher）是将明文消息编码后的序列划分成固定大小的组，每组明文分别在密钥的控制下变成等长的密文序列。这里我们主要考虑明文编码为二进制的情况。



分组密码的基本原理

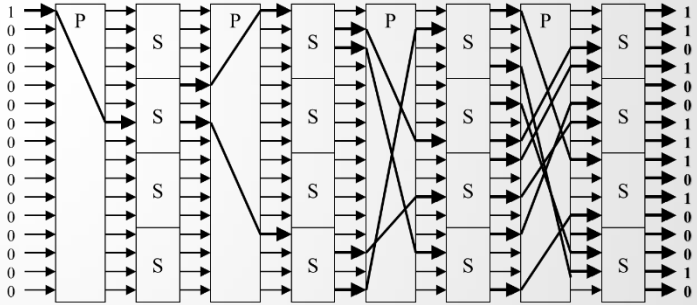
- **扩散（diffusion)**和**混淆（confusion)**是Shannon提出的设计密码体制的两种基本方法，其目的是为了抵抗攻击者对密码体制的统计分析。
  - **扩散**就是让明文中的每一位以及密钥中的每一位能够影响密文中的许多位，或者说让密文中的每一位受明文和密钥中的许多位的影响。这样可以隐蔽明文的统计特性，从而增加密码的安全性。当然，理想的情况是让明文中的每一位影响密文中的所有位，或者说让密文中的每一位受明文、密钥中所有位的影响.

分组密码的基本原理

- **扩散（diffusion)**和**混淆（confusion)**是Shannon提出的设计密码体制的两种基本方法，其目的是为了抵抗攻击者对密码体制的统计分析。
  - **扩散**
  - **混淆**就是将密文与明文、密钥之间的统计关系变得尽可能复杂，使得对手即使获取了关于密文的一些统计特性，也无法推测密钥。使用复杂的非线性代替变换可以达到比较好的混淆效果。

分组密码的基本原理

- **乘积密码体制**
  - 实际上，乘积密码就是扩散和混淆两种基本密码操作的组合变换，这样能够产生比各自单独使用时更强大的密码系统。选择某些较简单的受密钥控制的密码变换，通过乘积和迭代可以取得比较好的扩散和混淆效果。



SP网络

数据加密标准DES

- ✓ 1977年1月，美国政府宣布：将IBM公司设计的方案作为非机密数据的正式**数据加密标准（Data Encryption Standard，DES）**。
- ✓ DES是第一个广泛用于商用数据保密的密码算法，其分组长度为64位，密钥长度也为64位（其中有8位奇偶校验位，故实际密钥长度为56位）。
- ✓ 尽管DES目前因密钥空间的限制，已经被高级加密标准AES取代，但其设计思想仍有重要的参考价值。

数据加密标准(DES)

- ✓ 美国国家标准化局(NBS)于1977年公布
- ✓ 是Feistel 密码体系(FCS，由Horst Feistel发明)的一个具体的实现
- ✓ 对称的加密和解密结构
- ✓ 使用四个基本运算: 异或, 置换, 替换和循环位移
- ✓ 从70年代中期到2000年早期被广泛使用
- ✓ 逐步被AES 和其他更好的算法所取代

Feistel 密码体系(FCS)

- ✓ 将M 分成$2l$-bits长的分组 (必要时需要对最后一个分组填充)
- ✓ 仅使用异或和替换运算
- ✓ 从密钥K中生成n个固定长度子密钥: $K_1,\ldots,K_n$
- ✓ 将一个$2l$-bit 的输入分组分为两个部分: L_0 和 R_0, 长度均为l (分别表示分组的前缀和后缀)
- ✓ 对一个l-bit的输入串执行一个替换函数F 和一个子密钥得到一个l-bit的输出
- ✓ 加密和解密均执行n轮相同的序列的运算

Feistel密码结构

- **Feistel结构的具体实现依赖于以下参数和特征。**
  - 分组长度：分组长度越长意味着安全性越高（其他数据不变），但是会降低加密和解密的速度。这种安全性的增加来自更好的扩散性。传统上，64位的分组长度比较合理，在分组密码设计里很常用。然而，高级加密标准使用的是128位的分组长度。
  - 密钥长度：密钥长度较长同样意味着安全性较高，但会降低加密和解密的速度。这种安全性的增加来自更好的抗穷尽攻击能力和更好的混淆性。现在一般认为64位的密钥还不够，通常使用的密钥长度是128位。
  - 迭代轮数：Feistel密码的本质在于单轮不能提供足够的安全性，而多轮加密可取得很高的安全性。迭代轮数的典型值是16。
  - 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，抗攻击能力就越强。

FCS 加密和解密

FCS 加密

- ✓ 令$M = L_0R_0$; 在第i轮执行下列运算, $i = 1, \ldots, n$:

$$L_i = R_{i-1}$$

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

- ✓ 令$L_{n+1} = R_n$, $R_{n+1} = L_n$ 且$C = L_{n+1}R_{n+1}$

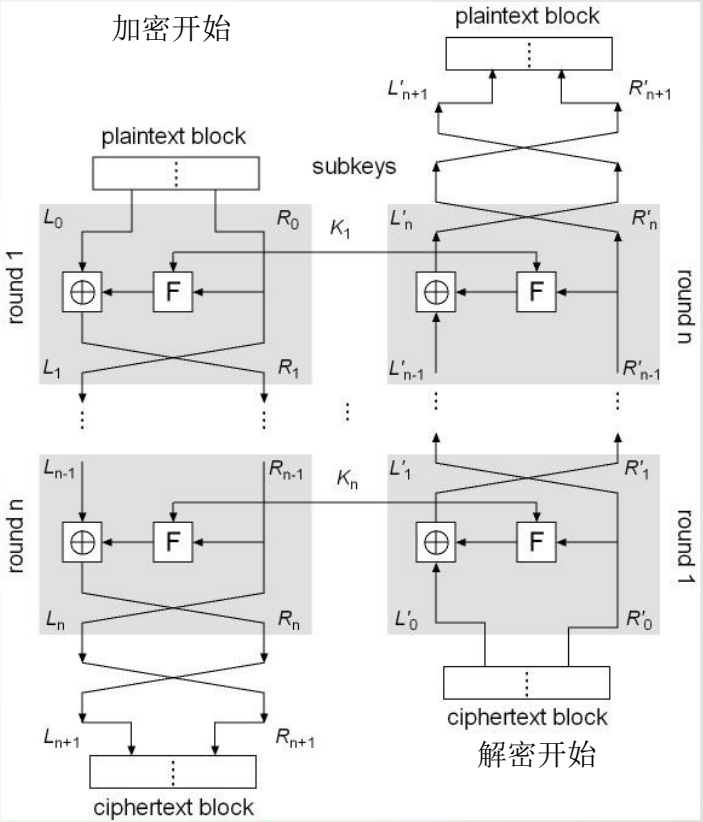
FCS 解密

- ✓ 与加密对称, 以相反的顺序使用子密钥
- ✓ 将C 重写为$C = L'_0R'_0$
- ✓ 在第i轮执行下列运算 ($i = 1, \ldots, n$):

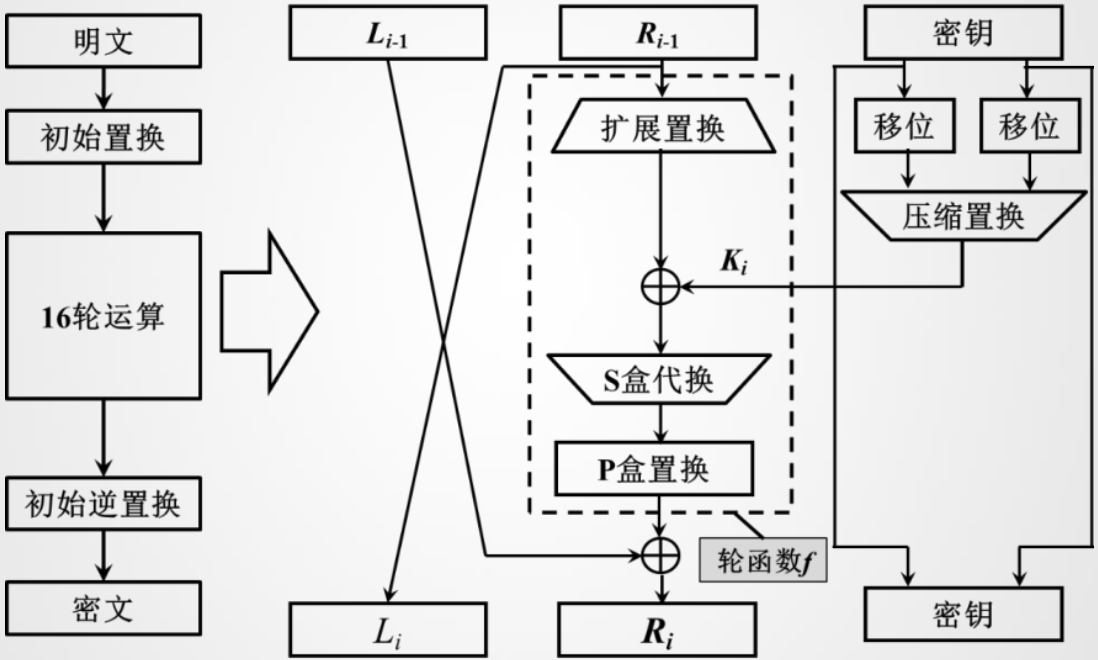
$$L'_i = R'_{i-1}$$

$$R'_i = L'_{i-1} \oplus F(R'_{i-1}, K'_{n-i+1})$$

- ✓ 另$L'_{n+1} = R'_n$, $R'_{n+1} = L'_n$
- ✓ 我们将得到$M = L'_{n+1}R'_{n+1}$



DES加密算法的结构流程



DES算法原理

DES采用了64位的分组长度和56位的密钥长度。除了初始置换和逆初始置换，DES结构与Feistel结构完全相同。DES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:

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

- 初始置换，　IP
- 16轮迭代，T_i ，　　i=1,2,...16
- 逆置换，IP^{-1}

DES 子密钥生成

- ✓ DES的分组大小为64 bits且加密密钥是 56 bits（由一个64-bit的串 $K = k_1\ k_2\ \ldots\ k_{64}$表示）
- ✓ DES用16个子密钥迭代16轮
- ✓ 子密钥生成:
  1. 从K中移除第$8i$-th 个bit ($i = 1, 2, \ldots, 8$)
  2. 对K的剩余56个bits中执行一个初始置换,记为$\mathrm{IP}_{\mathrm{key}}(K)$
  3. 将这个56-bit密钥分为两部分: U_0V_0, 均为28 bits
  4. 对U_0 和V_0 执行一个规定次数的左循环移位，得到U_iV_i:

$$U_i = \mathrm{LS}_{z(i)}\ (U_{i-1}), \qquad V_i = \mathrm{LS}_{z(i)}\ (V_{i-1})$$

  5. 用一个规定的压缩置换置换得到的U_iV_i, 生成一个48-bit的串作为子密钥, 记为 K_i

$$K_i = \mathrm{P}_{\mathrm{key}}\ (U_i\ V_i)$$

DES 替换盒

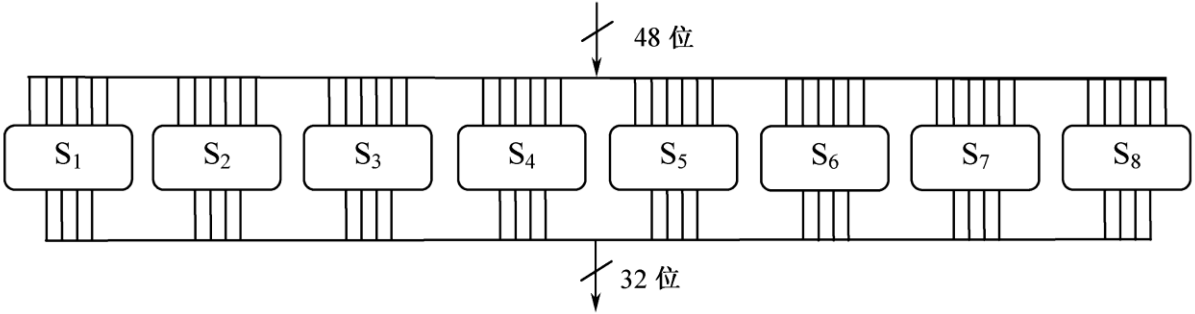
- DES的替换函数F定义如下:

$$F(R_{i-1}, K_i) = P(S(EP(R_{i-1}) \oplus K_i)), i = 1,\ldots,16$$

- 首先, 用$EP(R_i)$ 置换R_i 得到一个48-bit的串x
- 接着, 用48-bit子密钥K_i 异或x 得到一个48-bit串y
- 函数S 将y 转换为一个32-bits的串z, 利用8个4x16的特殊矩阵,即S-boxes
  - S-box中的每一项是一个4-bit串
  - 将y 分成8个分组,每个6-bits
  - 用第i^{th}个分组$b_1b_2b_3b_4b_5b_6$的第i^{th} 个矩阵
  - 令b_1b_6 为行号, $b_2b_3b_4b_5$ 为列号, 且返回对应的项
  - 每6-bit分组变换成一个4-bit串, 最终构成一个32-bit串z
- 最后, 用P置换z得到DES的F函数的结果
- 该结果, 与L_{i-1}异或, 就是R_i

S盒的设计

- S盒代换是DES算法中最重要的部分，也是最关键的步骤，因为其他的运算都是线性的，易于分析，只有S盒代换是非线性的，它比DES中任何一步都提供了更好的安全性。下图解释了S盒在函数F中的作用。代换函数由8个S盒组成，每个S盒输入6位，输出4位。



DES 加密步骤

- 重写IP(M) = L_0R_0, 这里$|L_0| = |R_0| = 32$
- 对$i = 1, 2, \ldots, 16$, 依次执行下列运算:

$$L_i = R_{i-1}$$

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

- 令$C = \text{IP}^{-1}(R_{16}L_{16})$

初始置换表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

逆初始置换表

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

- 这两个置换是互逆的。对于64位的二进制分组数据M，经过置换X=IP(M)后，再对它进行逆置换IP^{-1}，得到$Y = IP^{-1}(IP(M))$，就会恢复出M。例如，经过IP置换后，M的第1位被置换到第40位，再经过逆置换IP^{-1}，第40位又回到第1位的位置。

3DES/2, 2DES 和3DES/3

- DES的一个性质：
  - 没有两次加密与一次加密相同的情况: $E_K(M) \neq E_{K1}(E_{K2}(M)$
- 我们可以用多重DES
  - 用X 个密钥应用DES Y 次得到：YDES/X
  - 例如, 2DES/2, 3DES/2, 3DES/3
  - 我们可以用现有的DES有效的扩展加秘密钥的长度
  - 可抵御暴力攻击
- 例如, 3DES/2:

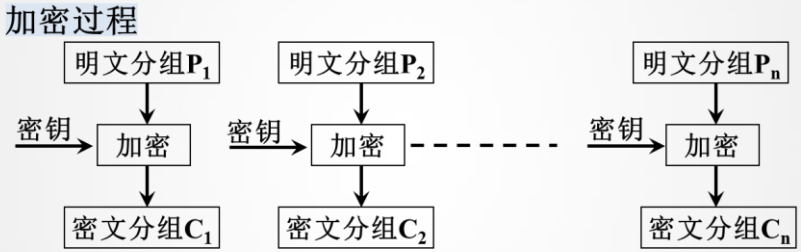
$$C = E_{K1}(D_{K2}(E_{K1}(M)))$$

$$M = D_{K1}(E_{K2}(D_{K1}(C)))$$

  - 注意: 其他组合 (如EEE和DDD)也是安全的,但利用现有的DES密文解密变得困难
  - 使用两个密钥将密钥扩展为112 bits,可使DES更安全从而抵御暴力攻击
- 注意2DES/2:
  - 2DES/2 使用与3DES/2一样多的密钥, 将密钥长度扩展为112
  - 然而, 2DES/2 无法抵御中途相遇攻击（*meet-in-the-middle* attack）

分组密码的工作模式

- 分组密码是将消息作为数据分组来加密或解密的，而实际应用中大多数消息的长度是不定的，数据格式也不同。当消息长度大于分组长度时，需要分成几个分组分别进行处理。为了能灵活地运用基本的分组密码算法，人们设计了不同的处理方式，称为**分组密码的工作模式**，也称为分组密码算法的运行模式。
- 分组模式能为密文组提供一些其他的性质，例如隐藏明文的统计特性、数据格式、控制错误传播等，以提高整体安全性，降低删除、重放、插入和伪造等攻击的机会。常用的工作模式有**电子编码本**模式、**密码分组链接**模式、**输出反馈**模式、**密码反馈**模式。



电子编码本模式(Electronic Code Book，ECB)

电子密码本 (ECB)

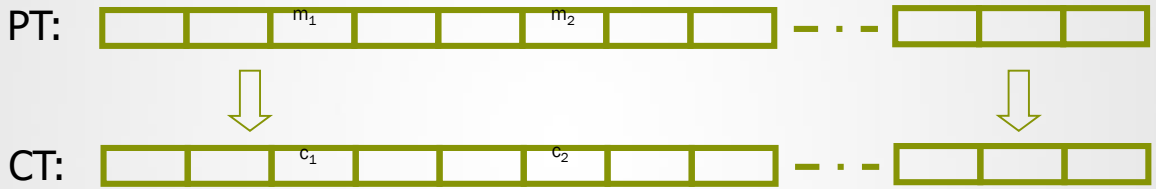
- ECB 独立的加密每个明文分组. 令C_i 为第i-th 密码本分组:

ECB 加密步骤	ECB 解密步骤
$C_i = E_k(M_i), \quad i = 1,2,\cdots,k$	$M_i = D_k(C_i), \quad i = 1,2,\cdots,k$

- 简单易懂. ECB常用于加密短的明文消息
- 然而,如果我们将我们的串进行分组, 两个分组相同是有可能的:
  $M_i = M_j \ (i \neq j)$
- 这使得攻击者能够得到关于加密的某些信息
- 其他工作模式以不同的方式处理这个问题

Incorrect use of block ciphers

Electronic Code Book (ECB):



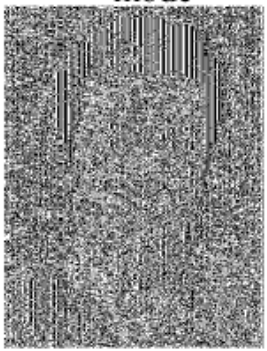
Problem:

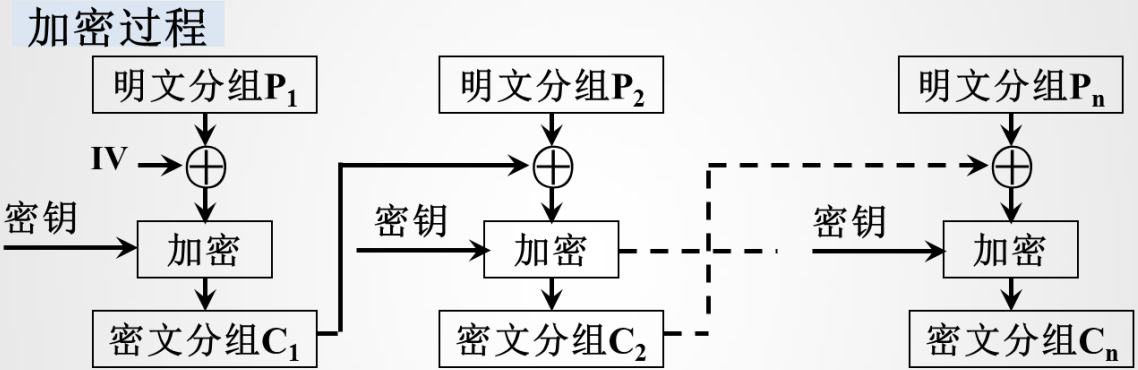
- ✓ if $m_1=m_2$ then $c_1=c_2$

In pictures

An example plaintext

Encrypted with AES in ECB mode





密码分组链接模式(Cipher Block Chaining，CBC)

密码分组链接 (CBC)

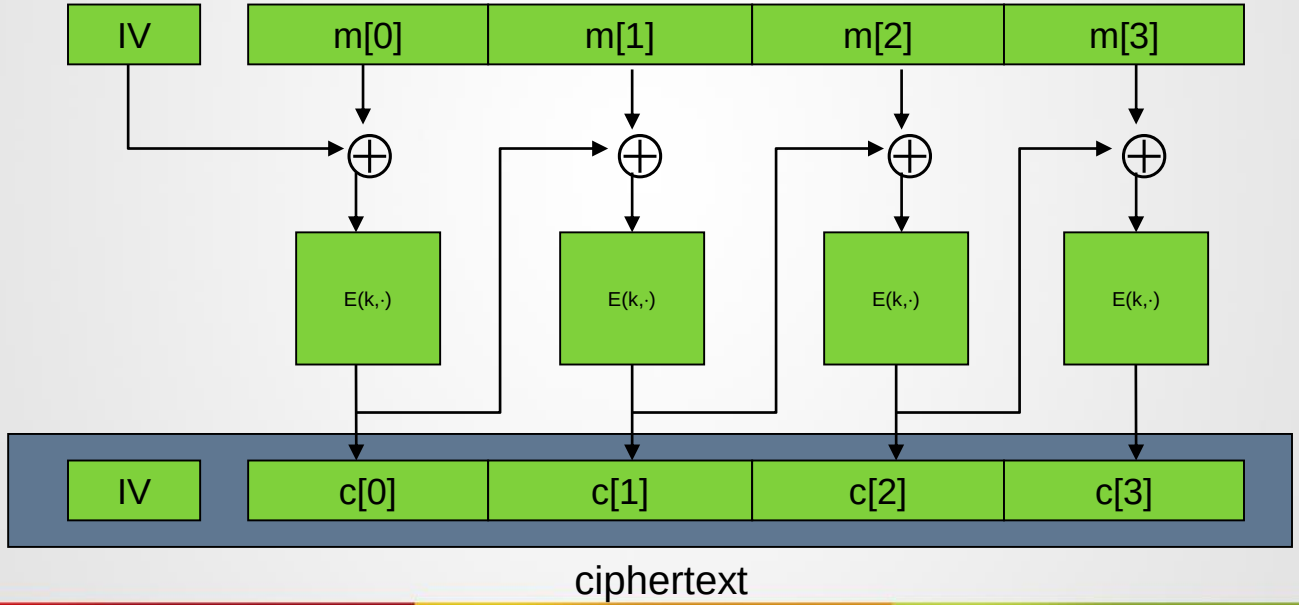
- 在ECB模式下，当明文消息 M 较长, 对于每个 $i \neq j$, $M_i = M_j$ *的概率 将增加*
- CBC 克服了ECB的缺点
- CBC模式中, 前一个密码本分组用于加密当前的明文分组
- CBC用一个初始的 l-bit 分组 C_0, 称为*初始向量*

CBC 加密步骤	*CBC* 解密步骤
$C_i = E_k(C_{i-1} \oplus M_i),$ $i = 1,2,\cdots,k$	$M_i = D_k(C_i) \oplus C_{i-1},$ $i = 1,2,\cdots,k$

- 如果在传输的时候，在密码本分组中出现一个比特错误怎么办？(扩散)
- 在 C_i 中的一个比特的变化影响接下来的分组

Correct use of block ciphers I: CBC mode

E a secure PRP. Cipher Block Chaining with random IV:

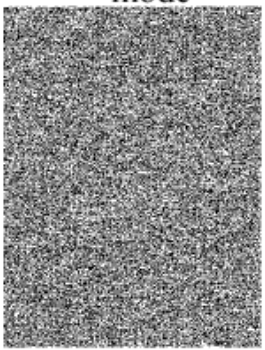


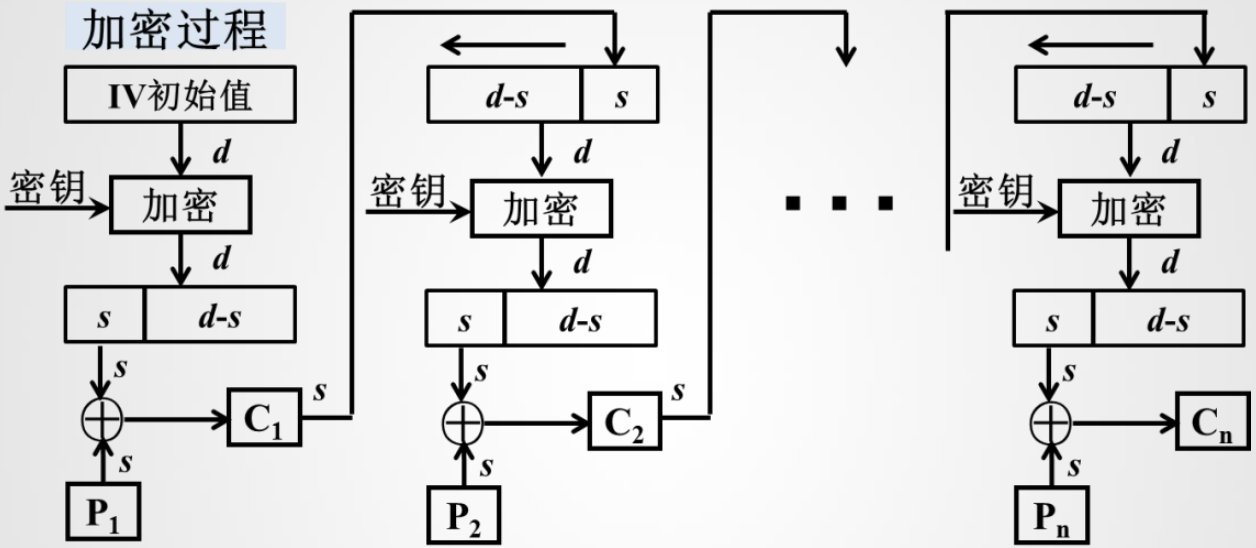
Q: how to do decryption?

In pictures

An example plaintext

Encrypted with AES in CBC mode





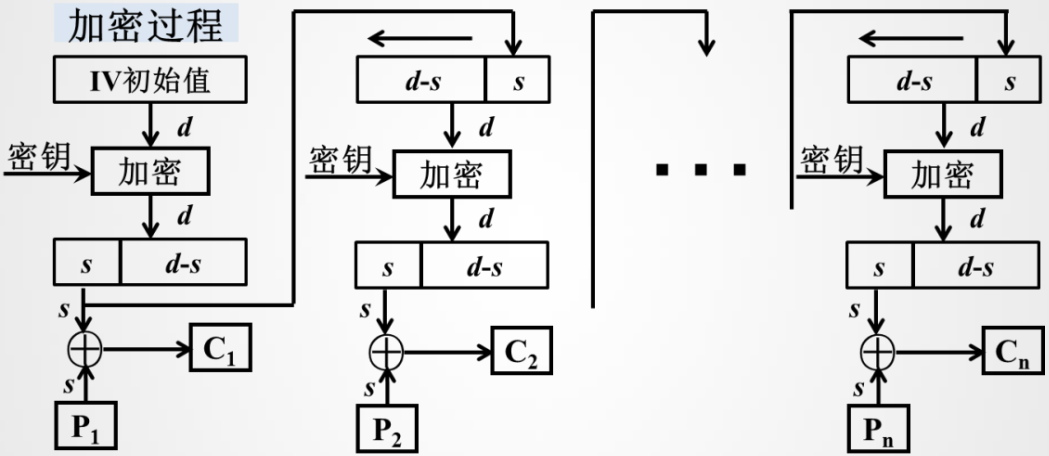
密码反馈模式 (Cipher FeedBack，CFB)

密文反馈 (CFB)

- CFB 把分组密码变为流密码
- M = $w_1 w_2 \ldots w_m$, 其中w_i的长度为s-bit
- 一次加密一个s-bit的分组（以明文字符为单位）：
  - s=8: ASCII码的流密码
  - s=16: unicode 流密码
- Also has an l-bit initial vector V_0

$$pfx_s(U) = \text{S bits prefix of U}$$
$$sfx_s(U) = \text{S bits subfix of U}$$

CFB 加密步骤	CFB 解密步骤
$U_i = E_k(V_{i-1})$	$U_i = E_k(V_{i-1})$
$C_i = w_i \oplus pfx_s(U_i)$	$w_i = C_i \oplus pfx_s(U_i)$
$V_i = sfx_{l-s}(V_{i-1})C_i$	$V_i = sfx_{l-s}(V_{i-1})C_i$
$i = 1,2,\cdots,m-1$	$i = 1,2,\cdots,m-1$
$U_m = E_k(V_{m-1})$	$U_m = E_k(V_{m-1})$
$C_m = w_m \oplus pfx_s(U_m)$	$w_m = C_m \oplus pfx_s(U_m)$



输出反馈模式 (Output FeedBack，OFB)

输出反馈 (OFB)

- OFB也把分组密码变为流密码
- CFB 和OFB仅有的区别是OFB 不在V_i设置C_i .
- 消息的反馈是独立的
- 在易出错的环境中使用

OFB 加密步骤	OFB 揭秘步骤
$U_i = E_k(V_{i-1})$ $C_i = w_i \oplus pfx_s(U_i)$ $V_i = sfx_{l-s}(V_{i-1})pfx_s(U_i)$ $i = 1,2,\cdots,m-1$ $U_m = E_k(V_{m-1})$ $C_m = w_m \oplus pfx_s(U_m)$	$U_i = E_k(V_{i-1})$ $w_i = C_i \oplus pfx_s(U_i)$ $V_i = sfx_{l-s}(V_{i-1})pfx_s(U_i)$ $i = 1,2,\cdots,m-1$ $U_m = E_k(V_{m-1})$ $w_m = C_m \oplus pfx_s(U_m)$

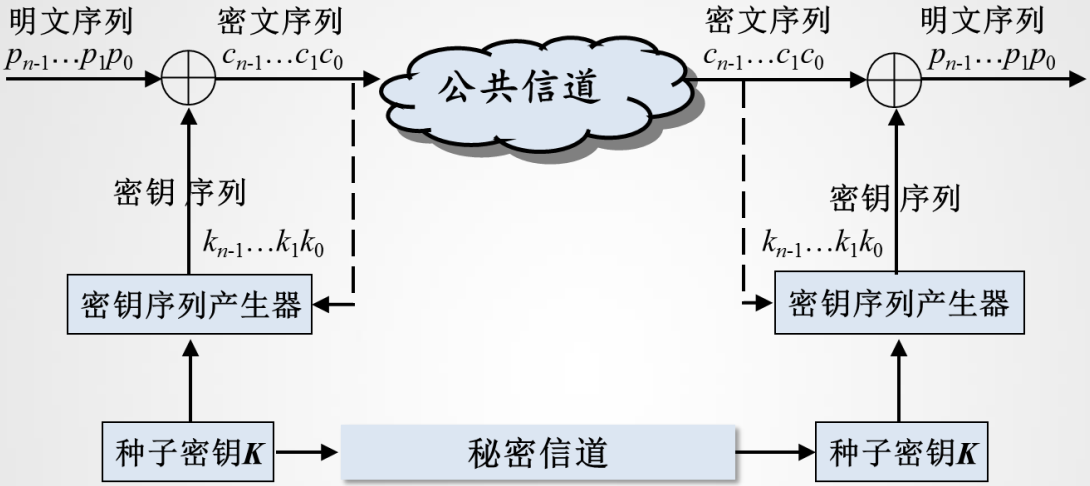
4.2.2 序列密码

- **序列密码（又称为流密码）**是一个重要的密码体制，也是手工和机械密码时代的主流密码。序列密码通常认为起源于20世纪20年代的Vernam密码。
- **序列密码属于对称密码体制，与分组密码相比较：**分组密码把明文分成相对比较大的块，对于每块使用相同的加密函数进行处理。分组密码是无记忆的。序列密码处理的明文长度为1比特，而且序列密码是有记忆的。序列密码又被称为状态密码，因为它的加密不仅与密钥和明文有关系，还和当前状态有关。两者区别不是绝对的，若把分组密码增加少量的记忆模块就形成了一种序列密码。

序列密码分类

- 序列密码通常划分为**同步序列密码**和**自同步序列**密码两大类。
  - 如果密钥序列的产生独立于明文消息，则此类序列密码为**同步序列密码**。在同步序列密码中，密（明）文符号是独立的，一个错误传输只会影响一个符号，不影响后面的符号。
  - 如果密钥序列的产生是密钥及固定大小的以往密文位的函数，则这种序列密码被称为**自同步序列密码**或**非同步序列密码**。自同步序列密码的优点是即使接收端和发送端不同步，只要接收端能连续地正确接收到n个密文符号，就能重新建立同步。

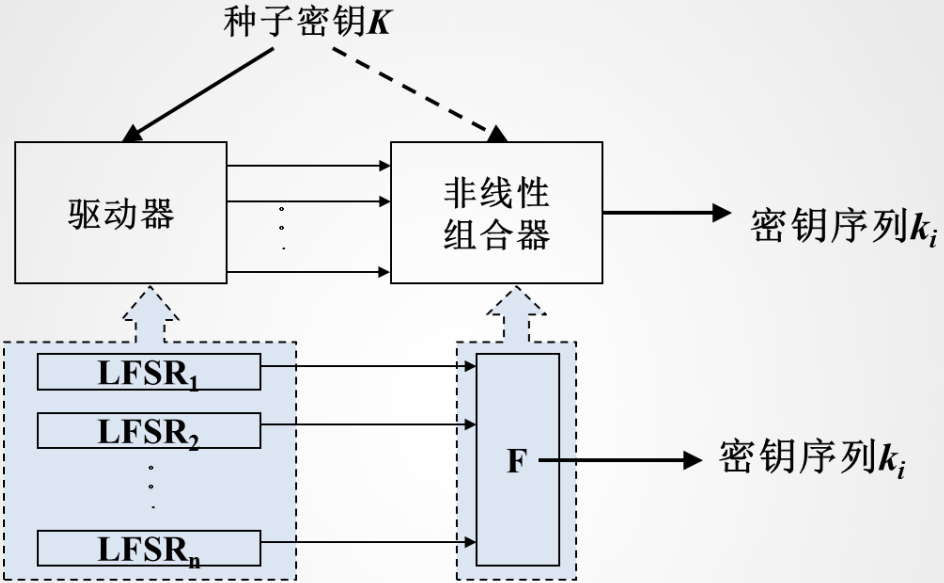
序列密码原理



- 序列密码是将明文划分成字符(如单个字母)，或其编码的基本单元（如0，1数字），字符分别与密钥序列作用进行加密，解密时以同步产生的同样的密钥序列实现。

密钥序列产生器

- ✓ 序列密码的安全强度主要依赖密钥序列的随机性，因此设计一个好的密钥序列产生器，使其**产生随机的密钥序列**是序列密码体制的关键。
- ✓ 密钥序列产生器的内部可将其分成两个部分——驱动部分和非线性组合部分（如图4-13），其中驱动部分产生控制生成器的状态序列，并控制生成器的周期和统计特性。非线性组合部分对驱动部分的各个输出序列进行非线性组合，控制和提高产生器输出序列的统计特性、线性复杂度和不可预测性等，从而保证输出密钥序列的安全强度。



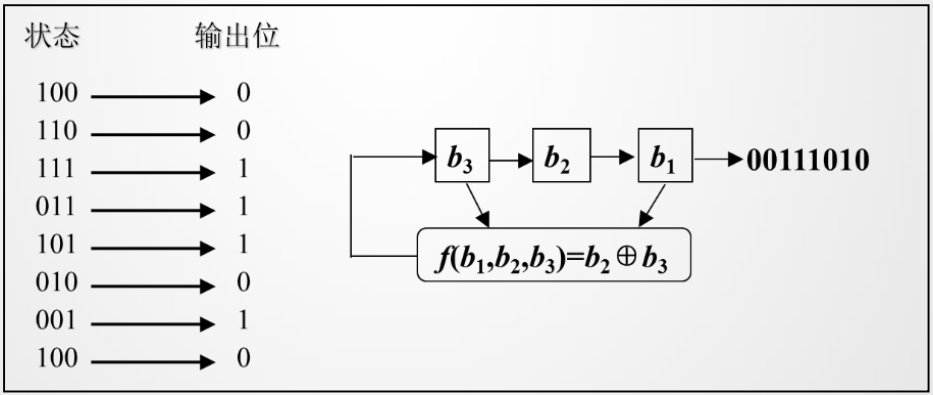
密钥序列产生器组成

线性反馈移位寄存器

- 序列密码的关键是设计一个随机性好的***密钥序列发生器***，为了研究密钥序列产生器，挪威政府的首席密码学家Ernst Selmer于1965年提出了移位寄存器理论，它是序列密码中研究随机密钥流的主要数学工具。尤其是线性反馈移位寄存器，因其实现简单、速度快、有较为成熟的理论等优点，而成为构造密码流生成器的最重要部件之一。

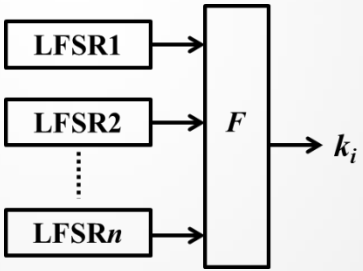
举例

- 一个3阶的线性反馈移位寄存器，反馈函数$f(b_1,b_2,b_3)=b_1 \oplus b_3$，初态为$(b_1b_2b_3)=100$，输出序列生成过程如下：



密钥序列生成器的组成

- 密钥序列生成器可分解为驱动部分和非线性组合部分，驱动子部分常用一个或多个LFSR实现，非线性组合子部分用非线性组合函数F实现。下面介绍第二部分：非线性组合子部分。

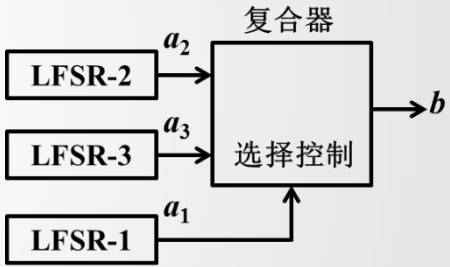


密钥序列生成器实例

- ✓ **Geffe发生器**
- ✓ Geffe发生器由两个LFSR作为复合器的输入，第三个LFSR控制复合器的输出。如果a_1、a_2和a_3是三个LFSR的输出，则Geffe发生器的输出表示为：

$$b = (a_1 \wedge a_2) \oplus (\neg a_1 \wedge a_3) = (a_1 \wedge a_2) \oplus (a_1 \wedge a_3) \oplus a_3 。$$

- ✓ 这个发生器的周期是三个LFSR周期的最小公倍数，它能实现序列周期的极大化，且0和1之间的分布大体是平衡的。



对称密码体制的局限性

- ✓ **密钥分配问题**。通信双方要进行加密通信，需要通过秘密的安全信道协商加密密钥，而这种安全信道可能很难实现。
- ✓ **密钥管理问题**。在有多个用户的网络中，任何两个用户之间都需要有共享的密钥，当网络中的用户n很大时，需要管理的密钥数目非常大。
- ✓ **难以实现不可否认功能**。当用户A收到用户B的消息时，无法向第三方证明此消息确实来源于B，也无法防止事后B否认发送过消息。

4.3 非对称密码体制

简介

- ✓ 非对称密码体制（asymmetric cryptosystems）为密码学的发展提供了新的理论和技术思想，是现代密码学最重要的发明，也可以说是密码学发展史上最伟大的革命。
- ✓ 一方面，非对称密码的算法是基于数学函数的，而不是建立在字符或位方式操作上的。
- ✓ 另一方面，与对称密码加、解密使用同一密钥不同，非对称密码使用两个独立的密钥，且加密密钥可以公开，因此又称为**公钥密码体制**。这两个密钥的使用对密钥的管理、认证都有重要的意义。

非对称加密体制模型

基本原理

- 1976年Diffie和Hellman在《密码学的新方向》一文中提出了公钥密码的思想，他们虽然没有给出一个真正的公钥密码算法，但首次提出了单向陷门函数的概念，将公钥密码体制的研究归结为**单向陷门函数**的设计，为公钥密码的研究指明了方向。

单向陷门函数

- 如果函数$f(x)$被称为**单向陷门函数**，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，若相应的x存在，则计算x使$y=f(x)$是容易的。

公钥密码体制的性质

- 利用公钥密码体制，通信双方无需事先交换密钥就可以进行保密通信。公钥密码体制可以提供以下功能：
  - **机密性（Confidentiality）**：通过数据加密来保证非授权人员不能获取机密信息。
  - **数据完整性（Data Integrity）**：通过数字签名来保证信息内容不被篡改或替换。
  - **认证（Authentication）**：通过数字签名来验证对方的真实身份。
  - **不可抵赖性（Nonrepudiation）**：通过数字签名来实现，使发送者不能事后否认他发送过消息，消息的接受者可以向第三方证实发送者确实发出了消息。

RSA公钥密码算法

- RSA密码是目前应用最广泛的公钥密码体制，该算法是由美国的Ron **R**ivest、Adi **S**hamir和Leonard **A**dleman三人于1978年提出的。
- 它既能用于加密，又能用于数字签名，易于理解和实现，是第一个安全、实用的公钥密码体制。
- RSA的基础是数论的欧拉定理，它的安全性依赖于大整数的因子分解的困难性。

4.4 HASH函数与消息认证

简介

- ✓ 随着网络应用的不断发展，信息安全除了要保障信息的机密性外，还要保障信息在存储、使用、传输过程中不被非法篡改，即信息的完整性。Hash函数可以将“任意长度”的输入经过变换以后得到固定长度的输出，也称为消息摘要。消息摘要能够用于完成消息的认证功能，消息认证是保证信息完整性的重要措施。
- ✓ Hash函数也称散列函数、哈希函数、杂凑函数等，是密码学的一个重要分支。Hash函数可以看作是一种单向密码体制，即它是一个从明文到密文的不可逆映射，即只有加密过程，不能解密。

Hash函数的基本概念

- ✓ Hash函数的单向特征和输出数据长度固定的特征使得它可以生成消息或其他数据块的“数据指纹”（也称消息摘要或Hash值），因此在消息认证和数字签名等领域有广泛的应用。
- ✓ 一般地，Hash值的生成过程可以表示为$h=H(M)$，其中M是“任意”长度的消息，H是Hash函数，h是固定长度的Hash值。

Hash函数的性质

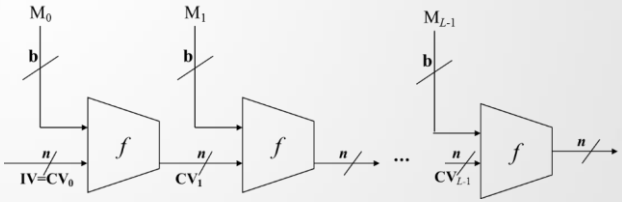
- ✓ **H可以用于“任意”长度的消息。** “任意”是指实际存在的。
- ✓ **H产生的Hash值是固定长度的。**这是Hash函数的基本性质。
- ✓ **对于任意给定的消息M，容易计算H(M)值。**这是要求Hash函数的可用性。
- ✓ **单向性（抗原像性）**：对于给定的Hash值*h*，要找到M使得H(M) = *h*在计算上是不可行的。

Hash函数的性质(续)

- **抗弱碰撞性（抗第二原像性）**：对于给定的消息M_1，要发现另一个消息M_2，满足$H(M_1) = H(M_2)$在计算上是不可行的。
- **抗强碰撞性**：找任意一对不同的消息M_1、M_2，使$H(M_1) = H(M_2)$在计算上是不可行的。
- **消息对应Hash值的每一比特应与消息的每一个比特有关联。**当消息原文发生改变时，求得的消息摘要必须相应的变化。

Hash函数的结构

- Hash函数的设计主要分为两类：一类是**基于加密体制实现的**，例如使用对称分组密码算法的CBC模式来产生Hash值；另一类是**直接构造复杂的非线性关系实现单向性**，后者是目前使用较多的设计方法。
- Hash函数的一般结构如图所示，称为迭代Hash函数结构。图中IV表示初始值，L为输入分组数，CV_i为链接变量，n为Hash值的长度，M_i为第i个输入分组，b是输入分组的长度，f是压缩函数。



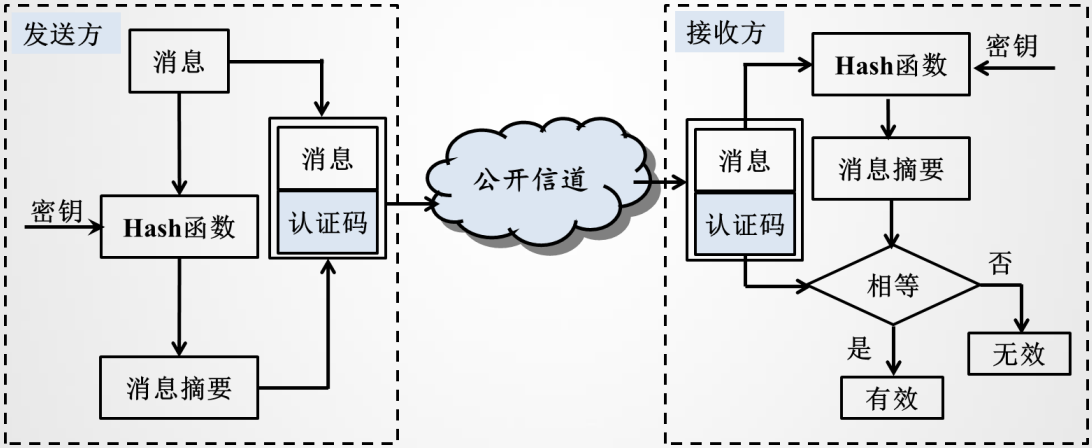
典型的Hash算法

- Hash算法中比较著名的是MD系列和SHA系列。
  - **MD系列**是在上个世纪90年代初由Mit laboratory for computer science和RSA data security inc的Rivest设计的，MD代表消息摘要（Message Digest），MD2(1989)、MD4(1990)和MD5(1991)都产生一个128位的信息摘要。
  - **SHA系列**算法是NIST根据Rivest设计的MD4和MD5开发的算法，国家安全局发布SHA作为美国政府标准，SHA（Secure Hash Algorithm）表示安全散列算法。

4.4.3 消息认证技术

- **消息认证的目的主要包括：验证信息来源的真实性和验证消息的完整性。**消息认证码（MAC，Messages Authentication Codes）是一种重要的消息认证技术，它利用消息和双方共享的密钥通过认证函数来生成一个固定长度的短数据块，并将该数据块附在消息后。
- **消息认证码是与密钥相关的Hash函数，也称消息鉴别码。**消息认证码与Hash函数类似，都具有单向性，此外消息认证码还包括一个密钥。不同的密钥会产生不同的Hash函数，这样就能在验证消息没有经过篡改的同时，验证是由哪一个发送者发送的。

消息认证码的实现过程


发送方
消息
密钥
Hash函数
消息
认证码
消息摘要
公开信道
接收方
Hash函数
密钥
消息
认证码
消息摘要
相等
否
是
无效
有效

MAC算法与加密算法

- ✓ 上述过程中，由于消息本身在发送过程中是明文形式，所以这一过程只提供认证性而未提供保密性。为提供保密性可在生成MAC之后或之前进行一次加密，而且加密密钥也需被收发双方共享。通常希望直接对明文进行认证，因此先计算MAC再加密的使用方式更为常用。
- ✓ MAC算法与加密算法类似，不同之处为MAC不必是可逆的（一般为多到一的映射），因此与加密算法相比更不易被攻破。
- ✓ 生成消息认证码的方法主要包括基于**加密函数的认证码**和**基于Hash的认证码**。

4.5 数字签名技术

- **数字签名(Digital Signature)**主要用于对数字消息进行签名，以防消息的冒名伪造或篡改，亦可以用于通信双方的身份鉴别。
- 数字签名具有**身份认证、数据完整性、不可否认性及匿名性**等方面的特点。随着计算机通信网络的迅速发展，特别是在大型网络安全通信中的密钥分配、认证及电子商务系统中，数字签名的使用越来越普遍，数字签名是防止信息欺诈行为的重要措施。

数字签名的特点

就签名的本质而言，需要具有以下特点：

- ✓ **不可否认性**：必须可以通过签名来验证*消息的发送者*、签名日期和时间。
- ✓ **不可抵赖性**：必须可以通过签名对所签署*消息的内容*进行认证。
- ✓ **可仲裁性**：必须可以由第三方通过验证签名来解决争端。

数字签名与手写签名的不同之处

- **首先，签名的对象不同。**手写签名的对象是纸质的文件，而数字签名的对象是传输在网络中的数字信息，是肉眼不可读的。
- **其次，实现的方法不同。**手写签名是将一串字符串附加在文件上，数字签名则是对整个消息进行某种运算。这一点在防篡改方面就凸显出数字签名的优势。数字签名与文件成为一个整体，任何改动都会对整个签名结果产生影响，从而免去了手写签名需要对文件的每一页进行手签的繁琐劳动。因此数字签名技术可以更有效地防止文件的篡改。

数字签名与手写签名的不同之处

- **再次，验证的方式不同。**手写签名的验证是通过和一个已有的签名进行对比，而模仿他人签名不是一件极其困难的事情，所以它的安全性得不到有效的保证。数字签名的验证则是通过一种公开的验证算法对签名进行计算，任何不一致都会被发现，因此具有很高的安全性。
- **最后，在保证机密性方面，数字签名比手写签名更具有优势。**因为数字签名可以实现对文件的加密，这样文件内容的机密性就得到了保证，而手工签名很难实现这一点。

数字签名应具有的功能

- ✓ 可以防范信息伪造。
- ✓ 防范信息篡改。
- ✓ 防范信息重放。
- ✓ 防止签名者抵赖。

数字签名与加密的不同之处

- 数字签名由公钥密码发展而来，与加密的不同之处在于：
  - 消息加密和解密可能是一次性的，它要求在解密之前是安全的；
  - 而一个签字的消息可能作为一个法律上的文件，如合同等，很可能在对消息签署多年之后才验证其签字，且可能需要多次验证此签字。

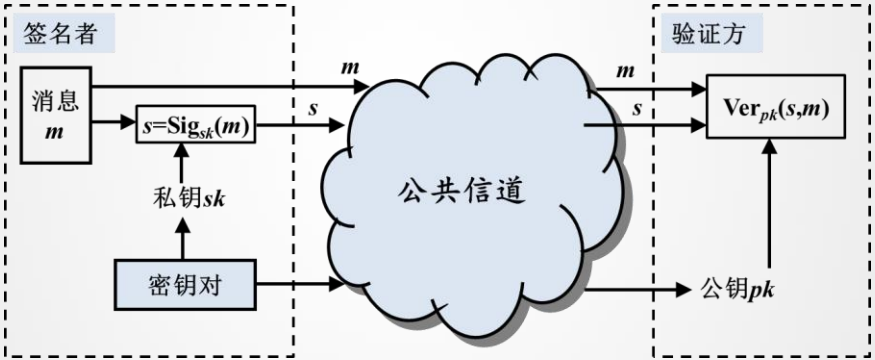
4.5.2 数字签名的原理

- 数字签名的目的是提供一种手段，使得一个实体把他的身份与某个信息捆绑在一起。一个消息的数字签名实际上是一个数，它依赖于签名者知道的某个秘密，也依赖于被签名信息的本身。
- 数字签名基于**两条基本的假设**：一是私钥是安全的，只有其拥有者才能获得；二是产生数字签名的惟一途径是使用私钥。

数字签名的组成

- 数字签名体制又称作数字签名方案，一般由两部分组成，即**签名算法**和**验证算法**。签名算法或签名密钥是由签名者秘密保有的，而验证算法或验证密钥应当公开，以方便他人进行验证。
- 一般来讲，数字签名方案包括三个过程：**系统的初始化过程**、**签名生成过程和签名验证过程。**

数字签名原理与过程



典型的数字签名体制

- 实现数字签名有很多种方法，基于对称密码体制，也可以依靠其共享密钥的保密性来实现数字签名，但其使用范围受到局限。目前数字签名多数还是利用公钥密码体制来设计的。典型的数字签名体制有：
  - 基于RSA的签名方案
  - DSA（Digital Signature Algorithm）签名体制

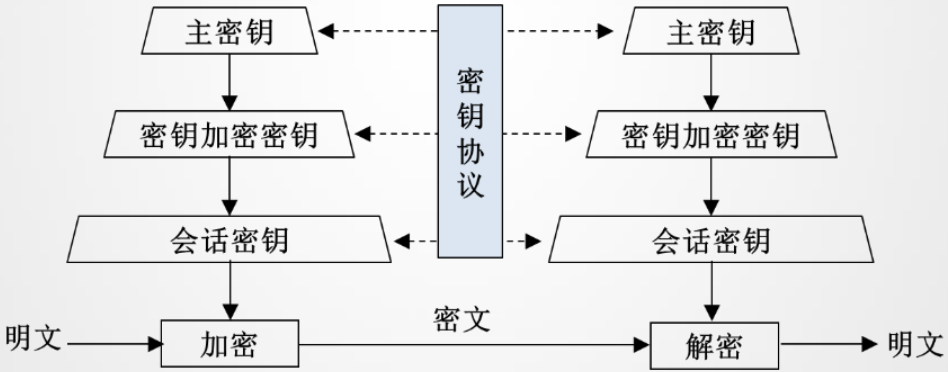
4.6 密钥管理技术

引言

- ✓ 现代密码体制要求密码算法是可以公开评估的，**整个密码系统的安全性并不取决对密码算法的保密或者是对密码设备等的保护，决定整个密码体制安全性的因素是密钥的保密性。**
- ✓ 密钥管理是密码学许多技术（如机密性、数据源认证、数据完整性和数据签名等）的基础，在整个密码系统中是极其重要的，密钥的管理水平直接决定了密码的应用水平。

4.6.1 密钥管理的层次结构

- 根据不同种类密钥所起的作用和重要性不同，现有的密码系统的设计大都采用了层次化的密钥结构，这种层次化结构与对系统的密钥控制关系是对应的。

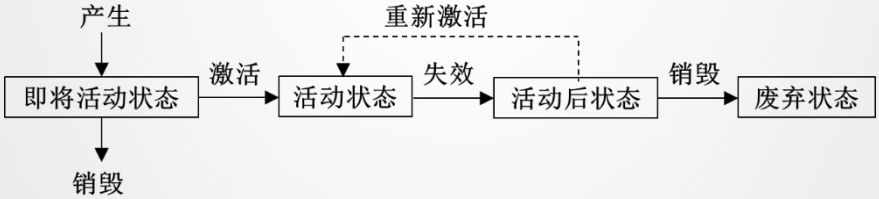


密钥分类

- 一般情况下，按照密钥的生存周期、功能和保密级别可以将密钥分为3类：会话密钥、密钥加密密钥和主密钥。
  - **会话密钥**：在一次通信或数据交换中，用户之间所使用的密钥，是由通信用户之间进行协商得到的。
  - **密钥加密密钥**：一般是用来对传输的会话密钥进行加密时采用的密钥，又称为次主密钥或者二级密钥。
  - **主密钥**：对应于层次化密钥结构中的最高层次，主密钥还起到标识用户的作用。

4.6.2 对称密码体制的密钥管理

- ✓ 在对称密码体制下，必须通过安全可靠的途径将密钥送至接收端，系统的保密性取决于密钥的安全性。
- ✓ 每个密钥都有其生命周期，有其自身的产生、使用和消亡的过程。在密钥的生命周期中有4个主要的状态：即将活动状态、活动状态、活动后状态和废弃状态。



会话密钥的建立方法

- 按照是否需要第三方可信机构来分，可分为无中心的密钥建立和有中心的密钥建立方式两类。
  - **无中心的密钥建立**是指用户直接将密钥传送给对方，此时参与者通常需要事先掌握一些资源。如果使用对称密码技术，在点对点的密钥建立过程中，要求在建立密钥之前参与协议的双方事先共享一个对称密钥，以便使用此共享的对称密钥作为密钥加密密钥来保护建立密钥时双方的通信。如果使用公钥密码技术，那么参与协议的双方也要事先知道对方的公钥。

无中心的密钥建立

该协议实现的前提是存在一种可交换的对称密码算法，即 $E_A(E_B(m)) = E_B(E_A(m))$

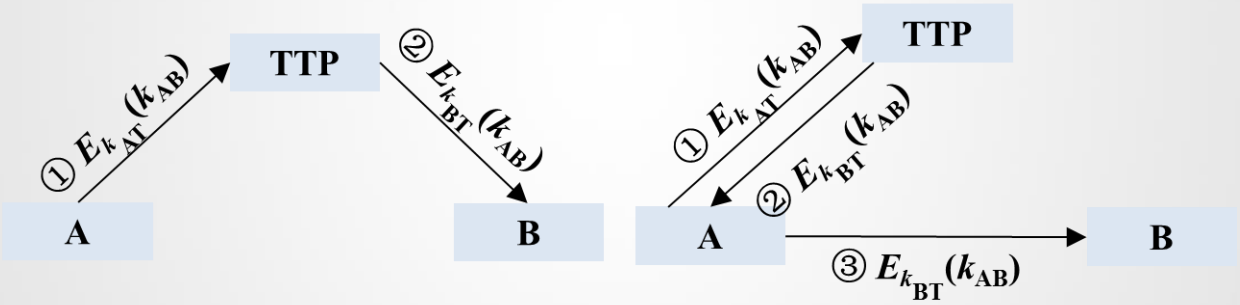
- ✓ A用自己的密钥加密k得到密文$c_1 = E_A(k)$，将密文 传送给B。
- ✓ B用自己的密钥加密c_1得到密文$c_2 = E_B(E_A(k))$，将密文c_2传送给A。
- ✓ A用自己的密钥解密c_2得到$c_3 = D_A(E_B(E_A(k))) = D_A(E_A(E_B(k))) = E_B(k)$，将$c_3$传送给B。
- ✓ B用自己的密钥解密c_3得到k。

注：虽然这个协议可以保证密钥的正确性，但是由于没有提供身份认证，很容易在执行过程中发生冒充行为。因此，在使用此协议时，需要有其他配套协议提供身份认证。

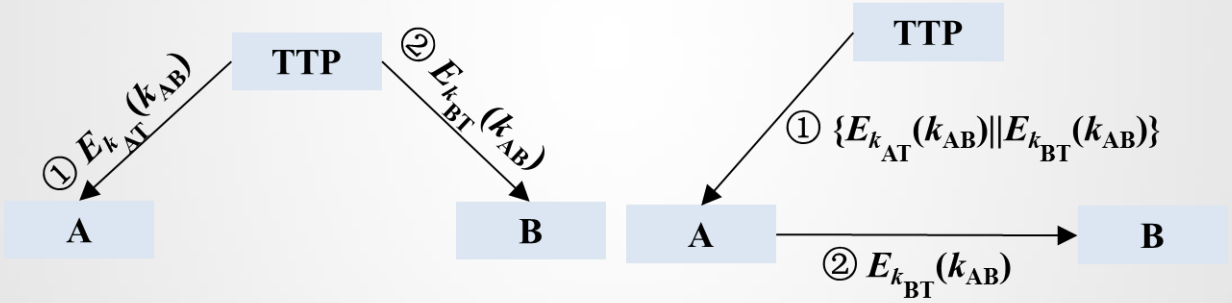
基于可信第三方的密钥建立

- 如果用户能和可信第三方（如密钥分配中心）之间建立了共享密钥，那么可以借助可信第三方的帮助，在任何两个互不认识的用户之间建立一个共享密钥。
- 设可信第三方TTP提供密钥的产生、密钥的鉴别、密钥的分发等服务。发送者A和接收者B分别与可信第三方TTP共享一个密钥，A与TTP的共享密钥为k_{AT}，B与TTP的共享密钥为 k_{BT}，A和B可以有两种途径建立密钥。

用户选择共享密钥的密钥建立过程



TTP选择共享密钥的密钥建立过程



4.6.3 非对称密码体制的密钥管理

- 在非对称密码系统中，公钥是公开的。公钥的这种公开性为信息安全通信带来了深远的影响，同时也为攻击者提供了可乘之机。
- 例如，攻击者可以用一个假公钥替换用户的真实公钥。因此，发展安全公钥密码系统的关键问题是如何确保公钥的真实性。
- 我们从**密钥协商**和**公钥证书**两个方面来讨论针对公钥密码系统的密钥管理方法和技术。

Diffie-Hellman密钥协商

设p是一个大素数，$g \in Z_p$是模p本原根，p和g公开，所有用户均可获取，并可为所有用户所共有。

(1)用户A随机选取一个大数a，$1 \le a \le p-1$，并计算$K_a \equiv g^a \pmod p$,并将结果传送给用户B。

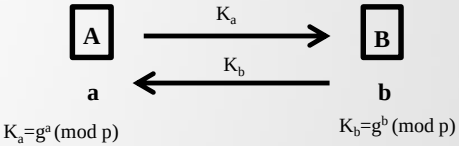
(2)用户B随机选取一个大数b，$1 \le b \le p-1$，然后计算$K_b \equiv g^b \pmod p$,并将结果传送给用户A。

(3)用户A计算$K \equiv (K_b)^a \pmod p$ 。

(4)用户B计算$K \equiv (K_a)^b \pmod p$ 。

用户A和用户B各自计算生成共同的会话密钥K。

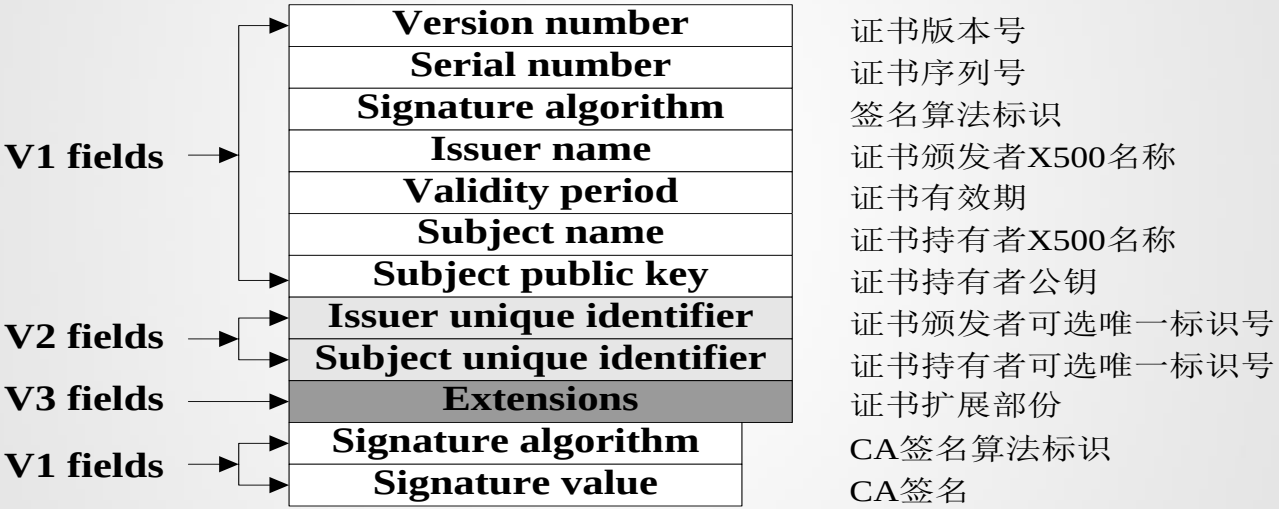
这是因为： $K \equiv (K_b)^a \equiv (g^b)^a \equiv g^{ab} \equiv (g^a)^b \equiv (K_a)^b \pmod p$ 。



公钥证书

- **公钥证书（Public Key Certificate）**是一种包含持证主体标识、持证主体公钥等信息，并由可信任的签证机构CA签署的信息集合。公钥证书主要用于确保公钥及其与用户绑定关系的安全。
- 公钥证书能以明文的形式进行存储和分配，任何一个用户只要知道可信任的签证机构*CA的公钥，就能验证证书的合法性*。如果验证正确，那么用户就可以相信该证书所携带的公钥是真实的，而且这个公钥就是证书所标识的那个主体的合法公钥。
- 存储在公钥证书中的最重要的信息有：证书持有者的标识、证书持有者的公钥、签证机构的标识、证书的序列号、证书的有效期、签证机构的签名等。

X.509证书结构


Version number
Serial number
Signature algorithm
Issuer name
Validity period
Subject name
Subject public key
Issuer unique identifier
Subject unique identifier
Extensions
Signature algorithm
Signature value
V1 fields
V2 fields
V3 fields
V1 fields
证书版本号
证书序列号
签名算法标识
证书颁发者X500名称
证书有效期
证书持有者X500名称
证书持有者公钥
证书颁发者可选唯一标识号
证书持有者可选唯一标识号
证书扩展部份
CA签名算法标识
CA签名

公钥管理技术的意义

- ✓ **公钥密码技术与对称密钥技术的最大区别**就是：用公钥技术加密消息，通信双方不需要事先通过共享的安全信道协商密钥。加密方只要得到接收方的公开密钥就可以加密消息，并将加密后的消息发送给接收方。由于公钥是公开的，因此需要一种机制来保证用户得到的公钥是正确的，即需要保证一个用户的公钥在发布的时候是真实的，在发布以后不会被恶意篡改。公钥管理技术为公钥的分发提供可信的保证。

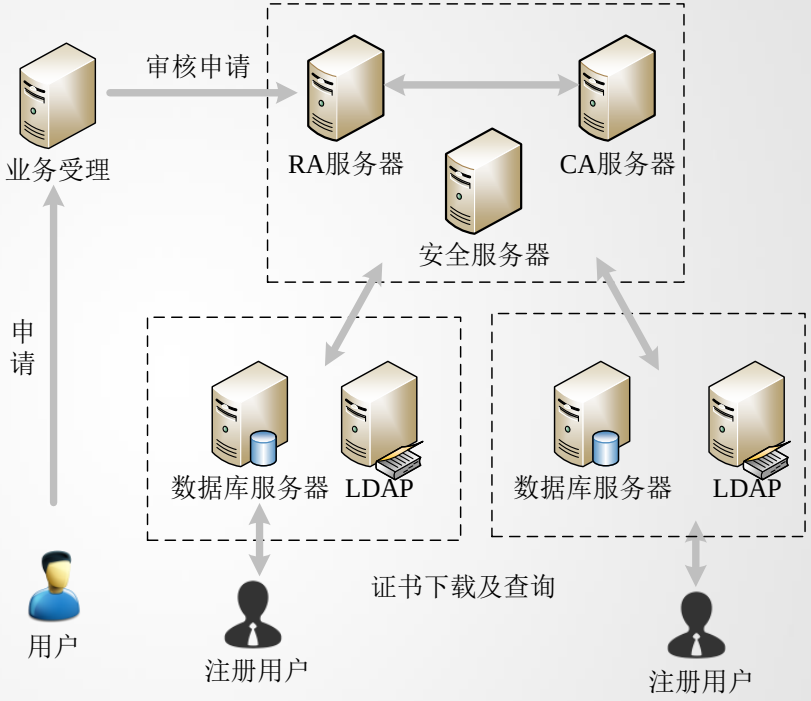
4.6.4 公钥基础设施技术（PKI）

- 公钥基础设施（Public Key Infrastructure，PKI）是网络安全的基础。其原理是利用非对称密码算法原理和技术所构建的，用来解决网络安全问题的一种普遍适用的基础设施。
- 有的学者把提供全面安全服务的基础设施，包括软件、硬件、人员和策略的集合称为PKI。
- PKI在网络信息空间的地位相当于电力基础设施在工业中的地位。可以说PKI是目前电子商务和电子政务所必不可少的安全基础。

PKI的一般结构

- 公钥基础设施（PKI）是一种遵循标准的密钥管理平台，涉及到多个实体之间的协作过程，主要包括：**认证中心**（Certificate Authority，CA）、**注册机构**（Registration Authority，RA）**证书数据库**（Certificate Database）、**密钥管理系统**（Key Manage System）、**证书撤销管理系统**（Certificate Revocation List Manage System）和**PKI应用接口系统**（PKI Application Interface System）及最终用户。

典型的PKI模型


审核申请
业务受理
RA服务器
CA服务器
安全服务器
申请
数据库服务器
LDAP
数据库服务器
LDAP
证书下载及查询
用户
注册用户
注册用户

本章小结

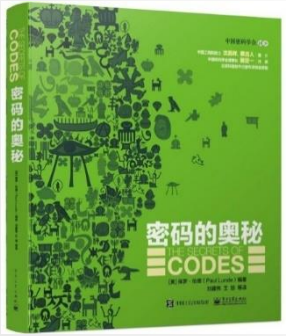
- ✓ 密码学是保障信息安全的核心，信息安全是密码学研究与发展的目标。保证数字信息机密性的最有效方法是使用密码算法对其进行加密；保证信息完整性的有效方法是利用Hash函数计算信息“指纹”，实现完整性检验；保证信息认证性的有效方法是密钥和Hash函数结合来确定信息的来源；保证信息不可抵赖性的有效方法是对信息进行数字签名。此外利用密码机制以及密钥管理技术可以有效地控制信息，使信息系统只为合法授权用户所用。

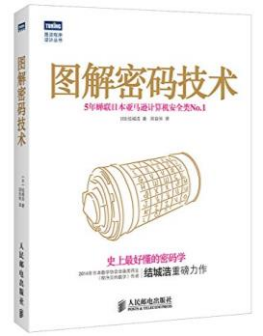
相关介绍

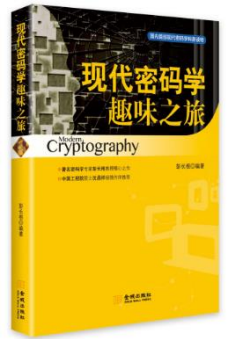
- ✓ http://www.cacrnet.org.cn/

中国密码学会
Chinese Association for Cryptologic Research.CACR

- ✓ 图书







作业

- ✓ 阅读第5章操作系统安全