第8章 网络安全

- 加密：防止信息落入非法用户之手
- 认证：在对话前确认对方的身份
- 认可（签名）：如何防止客户抵赖
- 完整性控制：如何确认你收到的信息在传输过程中没有被篡改

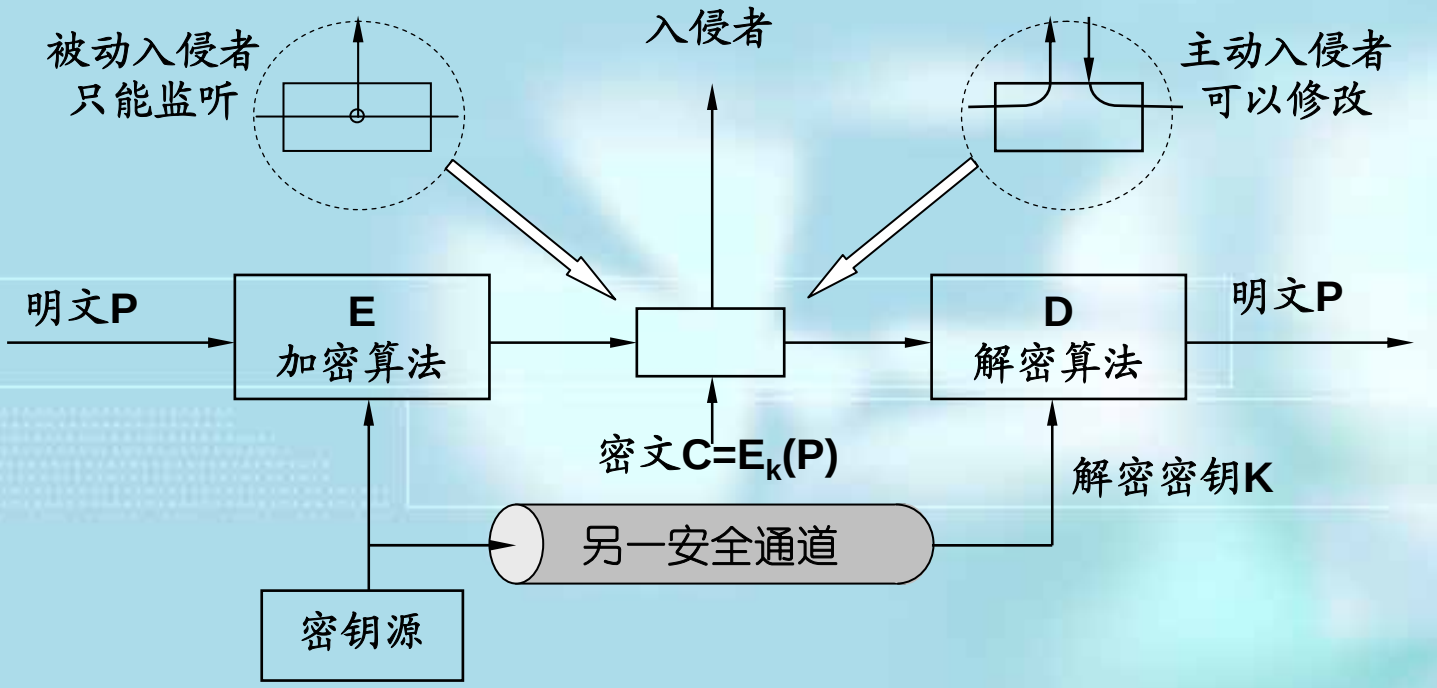
本章将讨论：

- 密码系统
- 对称密钥算法
- 公钥算法
- 数字签名
- 公钥管理
- 通信安全
- 认证协议
- **E-mail的安全**
- **Web安全**

密码系统

- 传统的数据加密
- 替换密码
- 变位密码

传统的数据加密模型



Tnbm P725 Fig. 8-2 加密模型（对称密钥）

传统的数据加密模型说明

- 明文P用加密算法E和加密密钥K加密，得到密文$C=E_K(P)$
- 在传送过程中可能出现密文截取者
- 到了接收端，利用解密算法D和解密密钥K，解出明文为 $D_K(C) = D_K(E_K(P)) = P$
- 截取者又称为攻击者，或入侵者
- 在这里我们假定加密密钥和解密密钥都是一样的，但实际上它们可以是不一样的（即使不一样，这两个密钥也必然有某种相关性）
- 密钥通常是由一个密钥源提供，当密钥需要向远地传送时，一定要通过另一个安全信道
- **Kerckhoff法则**：算法是公开的，密钥是保密的

密码系统

- 传统的数据加密
- 替换密码
- 变位密码

替换密码

- 凯撒密码

a-D、b-E、 c-F、d-G、e-H … … s-V … …、z-C

eg. 明文：**access control**

可变为： **DFFHVV FRQWURO**

密钥为：移**4**位

改进**1**：允许移位**k**位，**k**为密钥，解密要尝试**25**种可能

替换密码的再改进

- 用对照表

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

第二行的**26**个字母次序即为密钥

解密要尝试**26!**种情况。假设**1μs**试一种情况则需10^{13}年

但解密方法可用　**1**：分布式计算

2：用字频法

3：猜测字或短语

密码系统

- 传统的数据加密
- 替换密码
- 变位密码

变位密码

- 每个码不变，但位置改变，最常用的是列变位加密

M	E	G	A	B	U	C	K
7	4	5	1	2	8	3	6
p	l	e	a	s	e	t	r
a	n	s	f	e	r	o	n
e	m	i	l	l	i	o	n
d	o	l	l	a	r	s	t
o	m	y	s	w	i	s	s
b	a	n	k	a	c	c	o
u	n	t	s	i	x	t	w
o	t	w	o	a	b	c	d

Tnbm P729 Fig. 8-3 变位密码

例： 密钥为**MEGABUCK**

明文为:

pleasetransferonemillion
dollarstomyswissbankac
countsixtwotwo

密文为:

AFLLSKSOSELAWAIAT
OOSSCTCLNMOMANTE
SILYNTWRNNTSOWDPA
EDOBUOERIRICXB

本章将讨论：

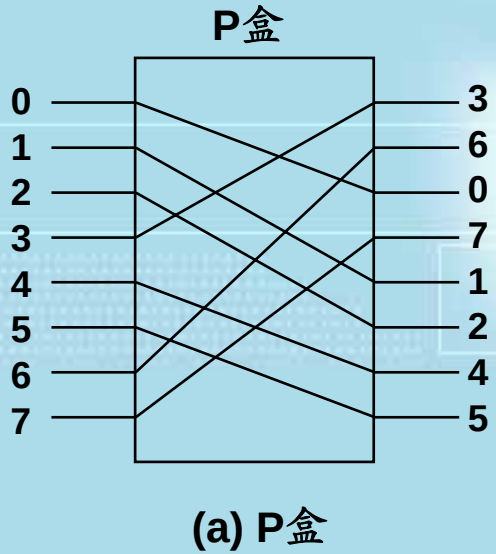
- 密码系统
- 对称密钥算法
- 公钥算法
- 数字签名
- 公钥管理
- 通信安全
- 认证协议
- **E-mail的安全**
- **Web安全**

对称密钥算法

- 变换盒P盒和替换盒S盒
- 乘积密码
- DES数据加密
- AES—高级加密标准
- 加密模式

变换盒：P盒

- P盒：实现变位



(a) P盒

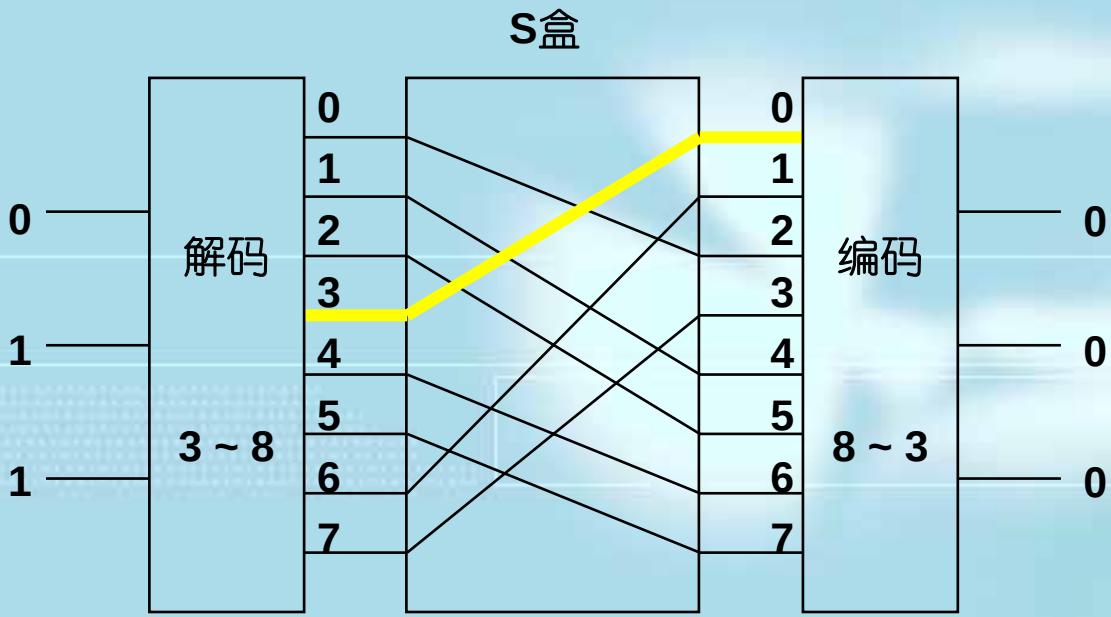
方式：用电路改变输入线的输出排列，图中列出8根线的变位，如这8位从上到下指定为**01234567**

则该P盒的输出为**36071245**

Tnbm P737 Fig. 8-6 乘积密码的基本元素

替换盒：S盒

- S盒：实现替换



(b) S盒

Tnbm P737 Fig. 8-6 乘积密码的基本元素

按图中的替换，如果8个八进制数01234567一个接一个地输入，那么输出序列将变为24506713，即2替换0，4替换1，注意n个比特的输入需要2^n条交换线

对称密钥算法

- 变换盒P盒和替换盒S盒
- 乘积密码
- DES数据加密
- AES—高级加密标准
- 加密模式

乘积密码

- 将一串盒子连接起来，组成乘积密码

(c) 乘积

Tnbm P737 Fig. 8-6 乘积密码的基本元素

乘积密码的实现

第一站对12根输入线作变换处理，从理论上讲，第二站可以为一个S盒，它把12比特数映射为另一个12比特数，但是，这样一个S盒的中段内需要$2^{12} = 4096$根跨接线，将12比特的输入分为4个3比特组，各组独立地进行替换处理，尽管这种方法没有通用性，但它却非常有效，在乘积密码中配置足够多的站，可以使输出成为输入的非常复杂的函数

对称密钥算法

- 变换盒P盒和替换盒S盒
- 乘积密码
- **DES数据加密**
- **AES—高级加密标准**
- 加密模式

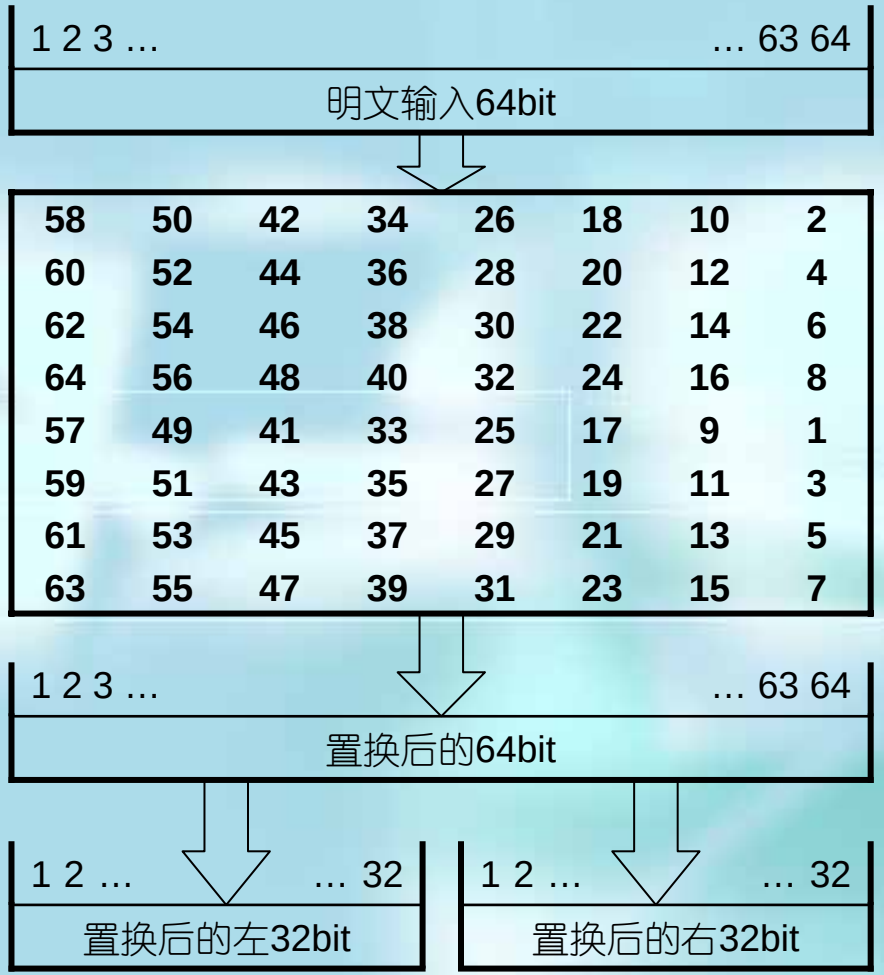
DES数据加密

- DES（Data Encryption Standard）数据加密标准
- 加密算法固定，根据不同的密钥产生不同的结果

明文按**64**比特块加密，生成**64 bit**的密文，此算法有一个**56 bit**的密钥作为参数（另加**8 bit**的奇偶位）

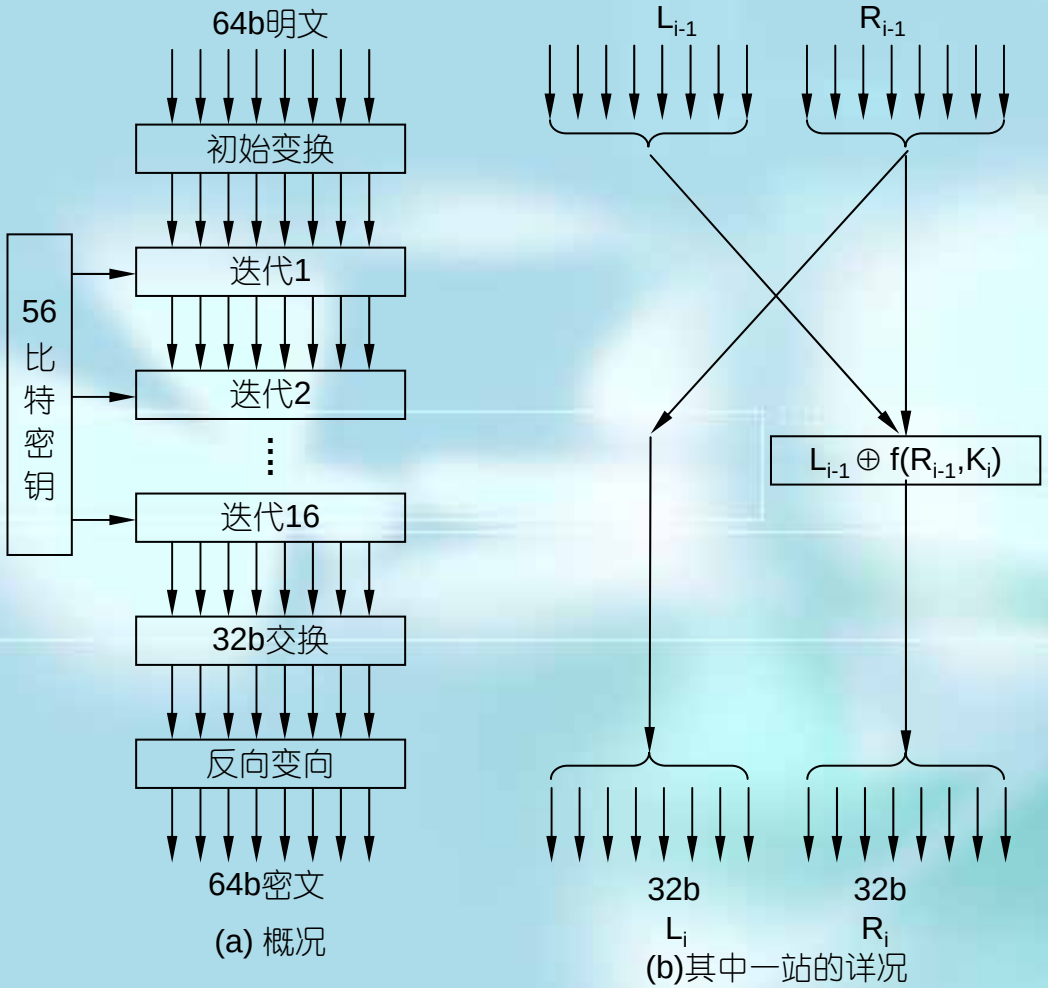
初始置换

- 即第一站，将**64 bit**明文作与密钥无关的变换，得到一个乱序的明文



16轮迭代的乘积变换

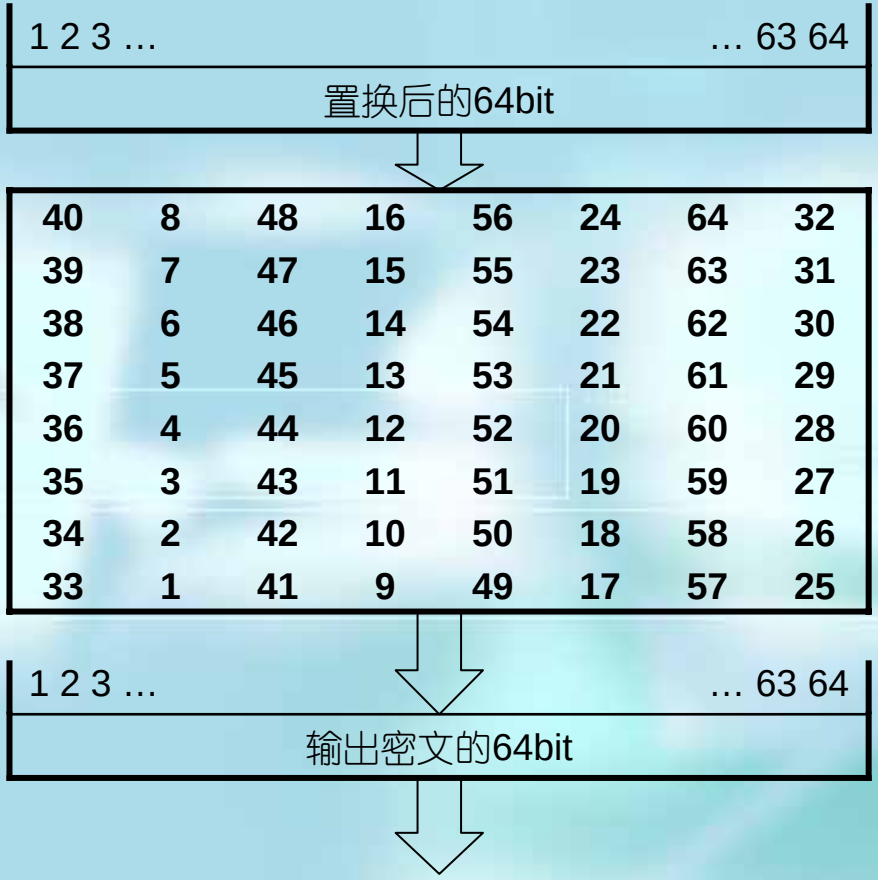
- 倒数第二站将左32 bit与右32 bit互换，余下的16站功能相同，但使用密钥的不同函数，解密用的密钥与加密密钥相同，只是解密步骤正好相反



Tnbm P739 Fig. 8-7 数据加密标准

逆初始置换

- 即最后一站是16轮迭代后的64 bit组进行变换，得到输出的密文组，是第一站变换的逆变换



其中函数 f 执行的步骤1、2

- 根据一个固定的变位和复制规则把32比特的R_{i-1}扩展成48比特的数E
- 把E与密钥K_i异或，并分成8组，每组6比特，分别送入8个不同的S盒

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

其中函数 f 执行的步骤3

- 每个S盒的64种可能的输入，将被映射为4比特的输出

8个S盒中某一个的6 bit输入映射为4 bit的输出的过程

X5	X4	X3	X2	X1	X0
1	0	1	0	1	1

X5	X0
1	1

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

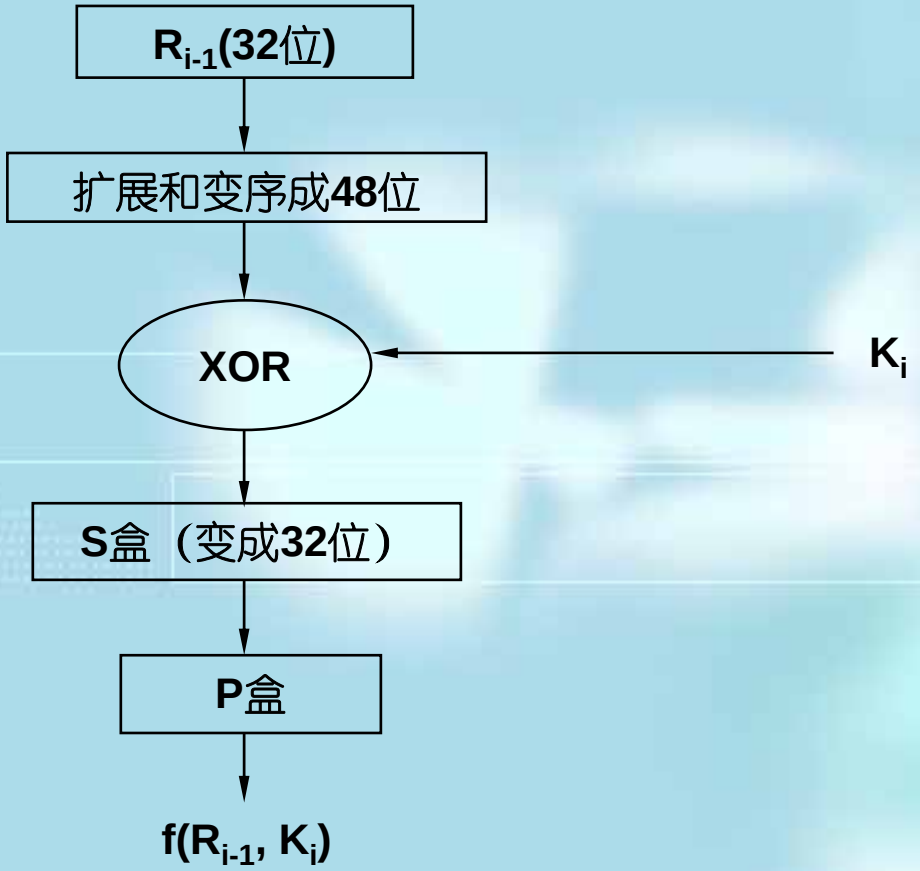
1	0	0	1

其中函数 f 执行的步骤4

- 8个S盒，每个S盒有4个输出，将通过一个32输入的P盒，再进行置换运算

置换后的32 bit输出将与左边的32 bit异或，作为下一轮迭代的右边数字段

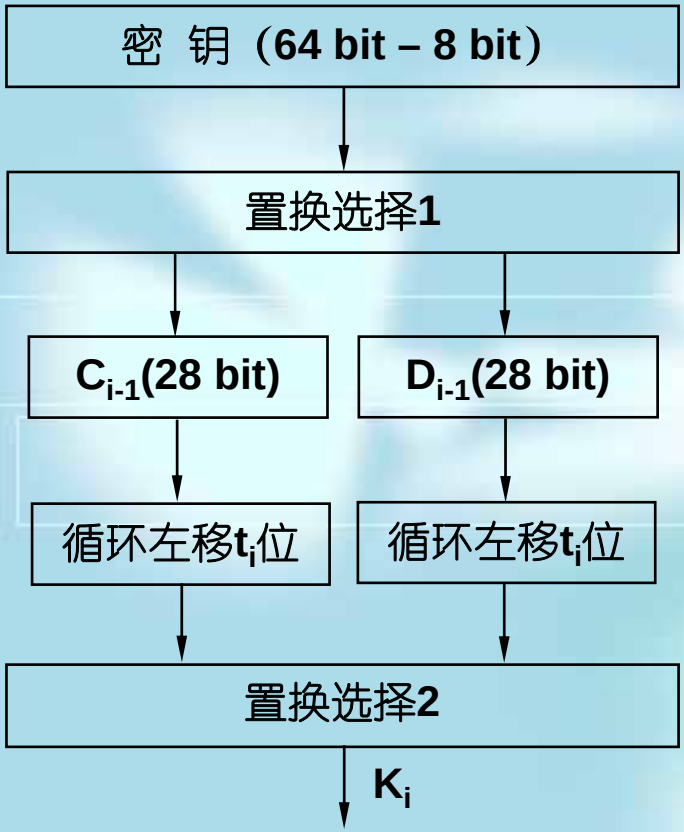
f 函数的计算如下



子密钥产生器K_i的计算

- 子密钥产生器框图
- 置换选择1
- 循环左移
- 置换选择2

子密钥产生器框图

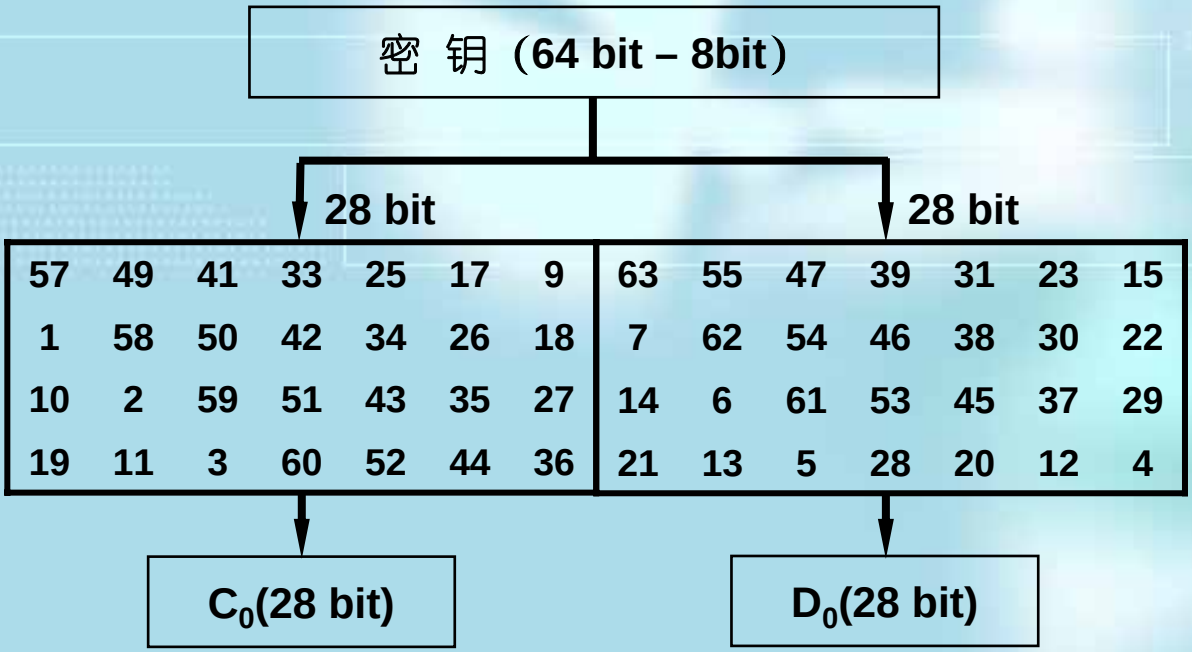

密 钥（64 bit – 8 bit）
置换选择1
C_{i-1}(28 bit)
D_{i-1}(28 bit)
循环左移t_i位
循环左移t_i位
置换选择2
K_i

子密钥产生器K_i的计算

- 子密钥产生器框图
- 置换选择1
- 循环左移
- 置换选择2

置换选择1

- 64 bit中的8、16、24、32、40、48、56、64位为校验位，其余56位为有效位，用于子密钥的计算



子密钥产生器K_i的计算

- 子密钥产生器框图
- 置换选择1
- 循环左移
- 置换选择2

循环左移

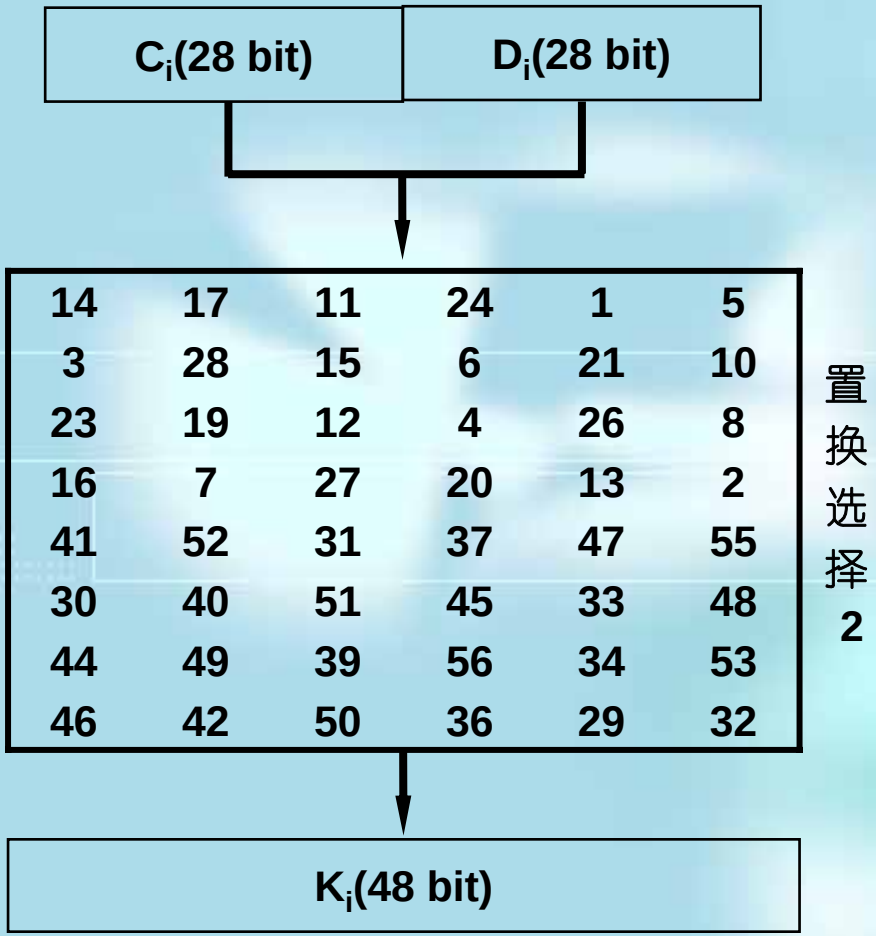
- 在各次迭代时，寄存器C和D的循环左移次数如下表:

第i次迭代	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
循环左移次数	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

子密钥产生器K_i的计算

- 子密钥产生器框图
- 置换选择1
- 循环左移
- 置换选择2

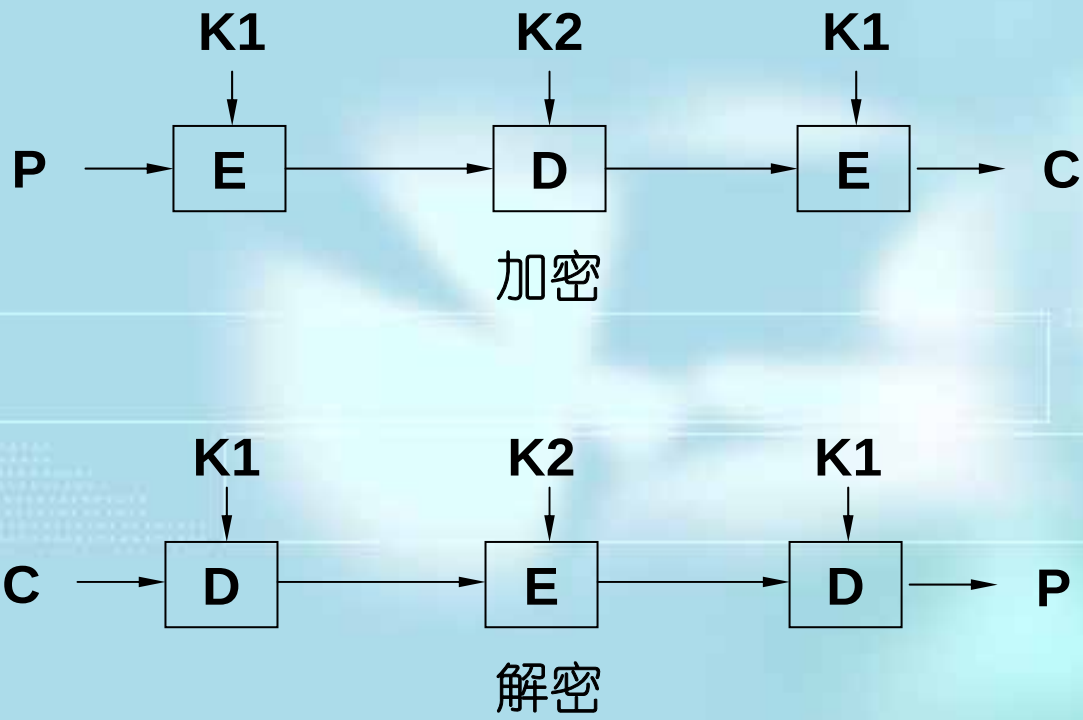
置换选择2


C_i(28 bit)
D_i(28 bit)
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
置换选择2
K_i(48 bit)

3重DES

- 解决DES算法中密钥太短的问题，由IBM公司提出
- 具体方法：用两个DES密钥、三个DES阶段来完成加密，首先，用K1对明文进行DES加密，然后用K2进行DES解密，最后再用K1进行DES加密，产生最终的密文
- 解密的方法正好相反

3重DES



Tnbm P741 Fig. 8-8 DES的三重加密和解密

对称密钥算法

- 变换盒P盒和替换盒S盒
- 乘积密码
- **DES**数据加密
- **AES—**高级加密标准
- 加密模式

AES—高级加密标准

- 用来替代DES标准，从全世界范围内征求方案
- AES要求：
  - 对称块加密
  - 设计是公开的
  - 必须支持128， 192和256三种密钥长度
  - 必须可以用软件和硬件实现
  - 算法必须是公开的，对所有人一视同仁

AES现有的方案--Rijndael

- 块长128位，密钥长128，192或256位
- 块长和密钥长度的选择是独立的，通常有128/128和128/256
- 与DES相同，该算法也由多次的替换和变位组成，迭代的次数取决于密钥长度和块长，128/128是十次，最多是14次
- 与DES不同的是所有的操作都是施加在所有的字节上

Rijndael算法

- 设明文与密钥长度相同，迭代次数为**R**，将明文和密钥看成**M**行**N**列的矩阵

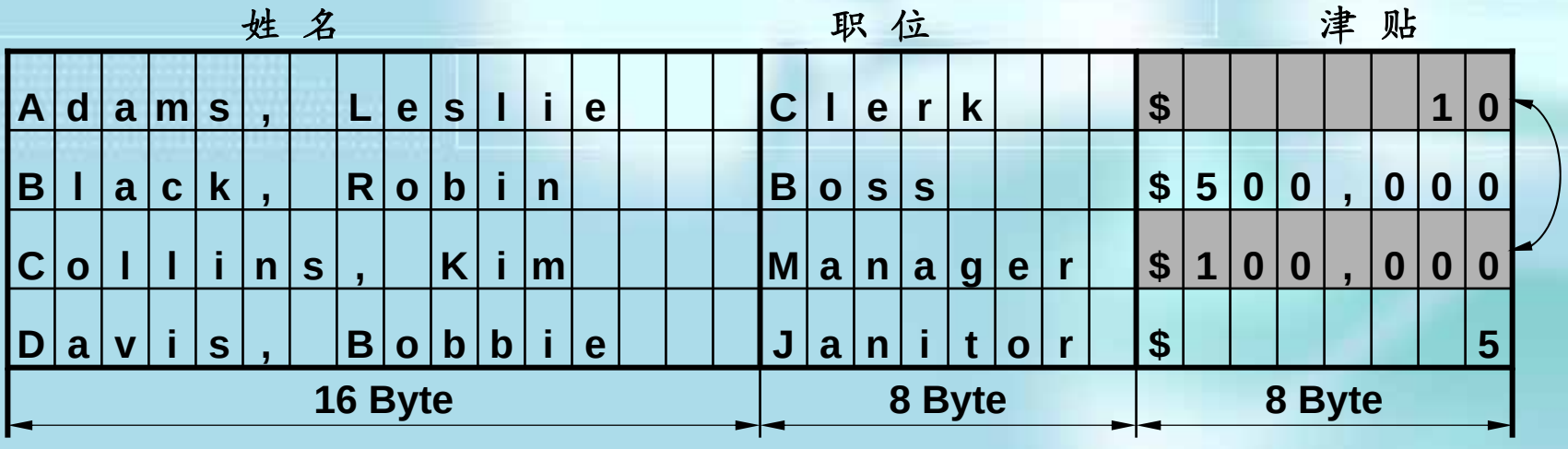
<table>
<tr><td colspan="2">从原始密钥产生R+1个与原始密钥等长的子密钥</td></tr>
<tr><td colspan="2">将第0个子密钥与明文作异或运算，得到第一次结果</td></tr>
<tr><td rowspan="4">循环K次</td><td>按字节对中间结果用一个S盒进行替换</td></tr>
<tr><td>按行进行左移操作，第J行左移J个字节</td></tr>
<tr><td>按列与一个常量矩阵相乘生成一个新列</td></tr>
<tr><td>将中间结果与第K个子密钥作异或运算生成下一个中间结果</td></tr>
</table>

对称密钥算法

- 变换盒P盒和替换盒S盒
- 乘积密码
- DES数据加密
- AES—高级加密标准
- 加密模式

加密模式1

- 电子代码本模式（**Electronic Code Book Mode**）：有一个明文和密文的对照本，同一明文总是对应同一密文，如**DES**，入侵者可以通过猜测解密，也可以通过分块特性篡改信息。如：



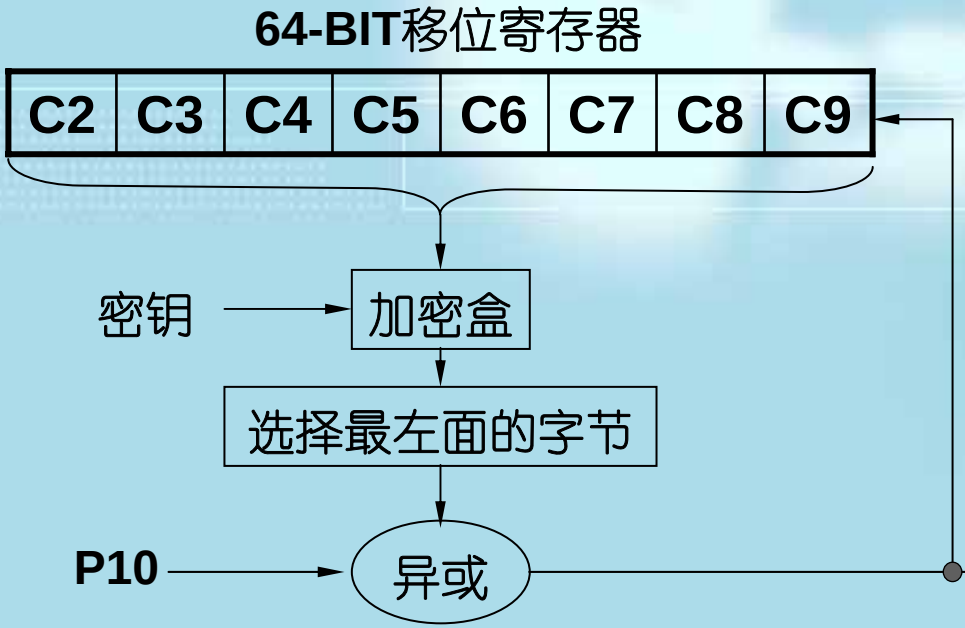
Tnbm P746 Fig. 8-11 用16个DES块加密的明文

加密模式2

- 密码块连接模式（**Cipher Block Chaining Mode**）：每个明文块在加密以前先与前一个密文块异或，第一块与一个随机选取的初始向量异或，这样同样的明文将不再映射到同样的密文了，如**DES**链就属于这种加密模式

加密模式3

- 密码反馈模式（**Cipher Feedback Mode**）：**DES**链的缺点在于解密前必须完整地收到**64**位的密文，**CFM**用于按字节加密

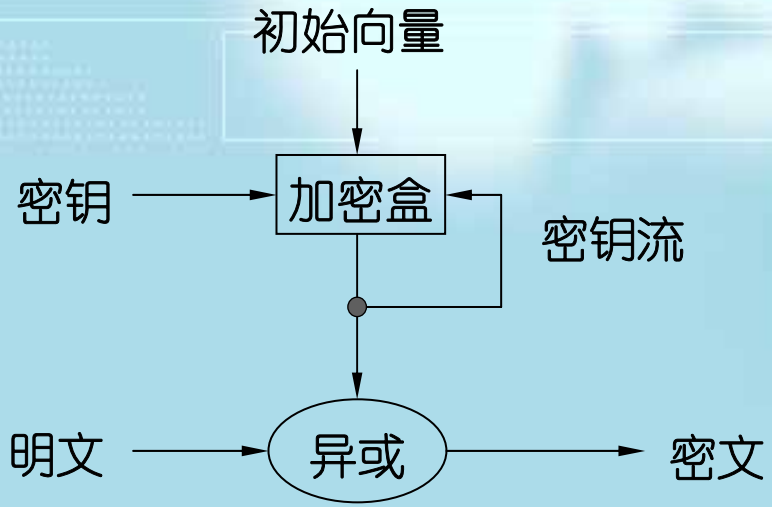


Tnbm P748 Fig. 8-13
密码反馈模式

(a) 加密

加密模式4

- 流加密模式（**Stream Cipher Mode**）：块加密中，1位传输错误将影响整个块，而流加密模式中，1位传输错误只影响1位



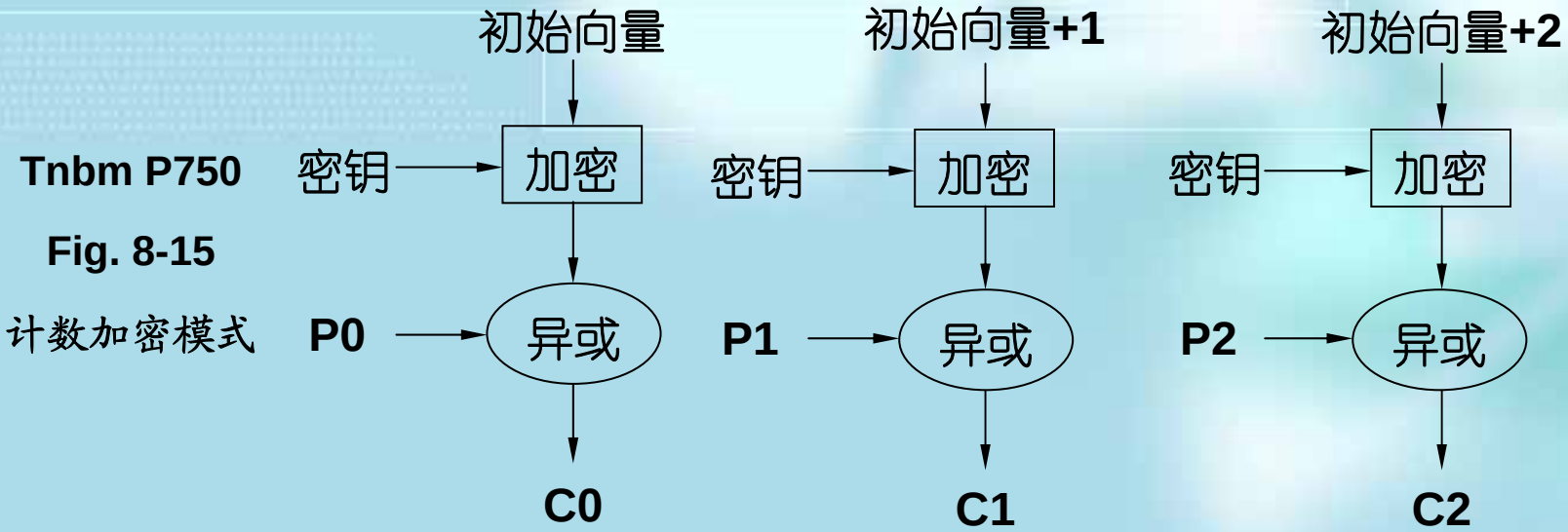
Tnbm P749 Fig. 8-14

流加密模式

(a) 加密

加密模式5

- 计数器模式（**Counter Mode**）：除了电子代码本模式外，上述的其他模式都有一个共同的问题—不能随机访问密文，必须顺序访问才能解密，在**CM**中，初始化向量加上计数器的值被加密，然后与明文块异或，结果为密文



Tnbm P750

Fig. 8-15

计数加密模式

本章将讨论：

- 密码系统
- 对称密钥算法
- 公钥算法
- 数字签名
- 公钥管理
- 通信安全
- 认证协议
- **E-mail的安全**
- **Web安全**

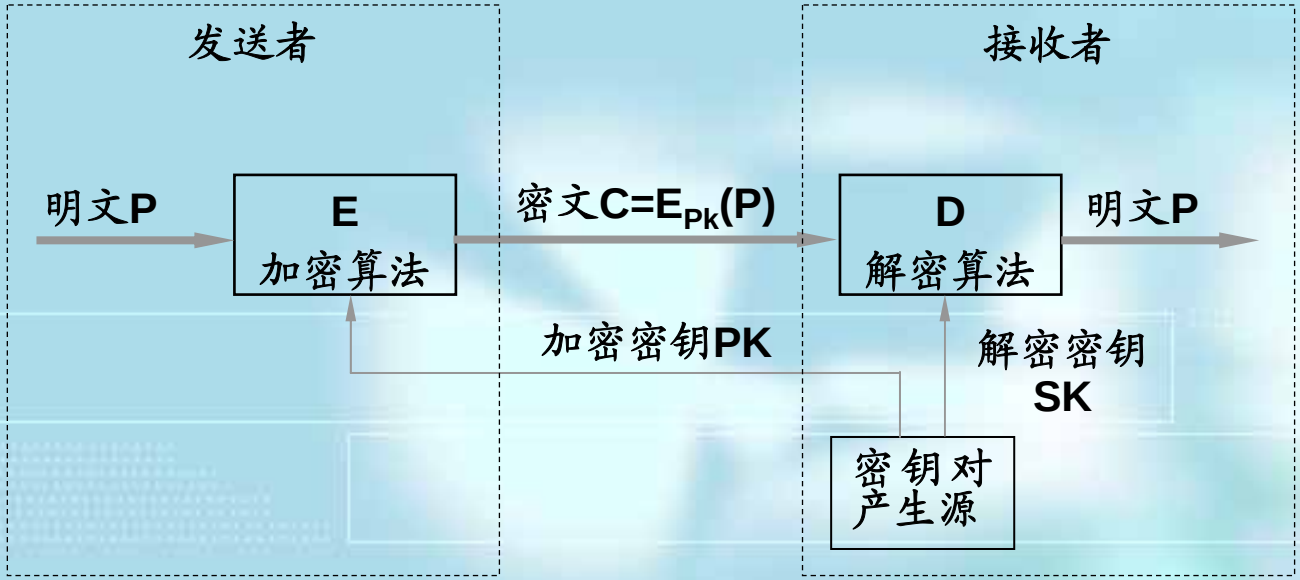
公开密钥算法

- 密钥是成对产生的
- 加密密钥不能用来解密

$$D_{SK}(E_{PK}(P)) = P \quad \text{但} D_{PK}(E_{PK}(P)) \neq P$$

- 加密密钥和算法是公开的，解密密钥是保密的
- 从PK(加密密钥)导出SK(解密密钥)极其困难

公开密钥算法模型



- 公开密钥算法中**RSA**算法最有代表性
- **RSA**算法：基于数论

密钥的选取

- 选择两个大质数，**p**和**q**（典型地应大于10^{100}）
- 计算**n = p * q**和**z =（p - 1）*（q - 1）**
- 选择一个与**z**互质的数**d , (d, n)**为解密密钥
- 找出**e,** 使**e * d（mod z）= 1 (e,n)**为加密密钥

公开密钥为（**e**、**n**），私有密钥为（**d**、**n**）

n为可编码的最大数

加密和解密算法

- 把明文看成一个bit串，并划分成每块k个bit，满足$2^k < n$，$P = 2^k$
  - 对原始信息P加密：

    使用公开密钥为（e、n），计算密文$C = P^e \pmod n$
  - 对加密信息C解密：

    使用私有密钥为（d、n），计算明文$P = C^d \pmod n$

加密和解密算法举例

- 选择p = 3，q = 11（实际中p、q为大质数）

  Tnbm P754

  n = p * q = 33，z =（p - 1）*（q - 1）= 20

  因为7与20互质，所以选择**d = 7**

  7e（mod 20）= 1的数有**21、41、61、81、101……**

  选**e = 3**

  对原始信息P加密：

  即计算密文$C = P^3$（mod 33）使用公开密钥为（3、33）

  对加密信息C解密：

  即计算明文$P = C^7$（mod 33）使用私有密钥为（7、33）

加密和解密算法举例

- $P = 2^k < 33$，$k = 5$即用5bit表示一个信息，有32种表示
- 分别用其中的1 – 26表示26个英文字母A - Z

如明文为SUZANNE可表示为19 21 26 01 14 14 05

明文(P)			密文(C)		解密后	
符号	数值	P^3	$P^3 \pmod{33}$	C^7	$C^7 \pmod{33}$	符号
S	19	6859	28	13492928512	19	S
U	21	9261	21	1801088541	21	U
Z	26	17576	20	1280000000	26	Z
A	01	1	1	1	1	A
N	14	2744	5	78125	14	N
N	14	2744	5	78125	14	N
E	05	125	26	8031810176	5	E

发送者的计算　　接收者的计算

Tnbm P754 FIG. 8-17 RSA算法举例

本章将讨论：

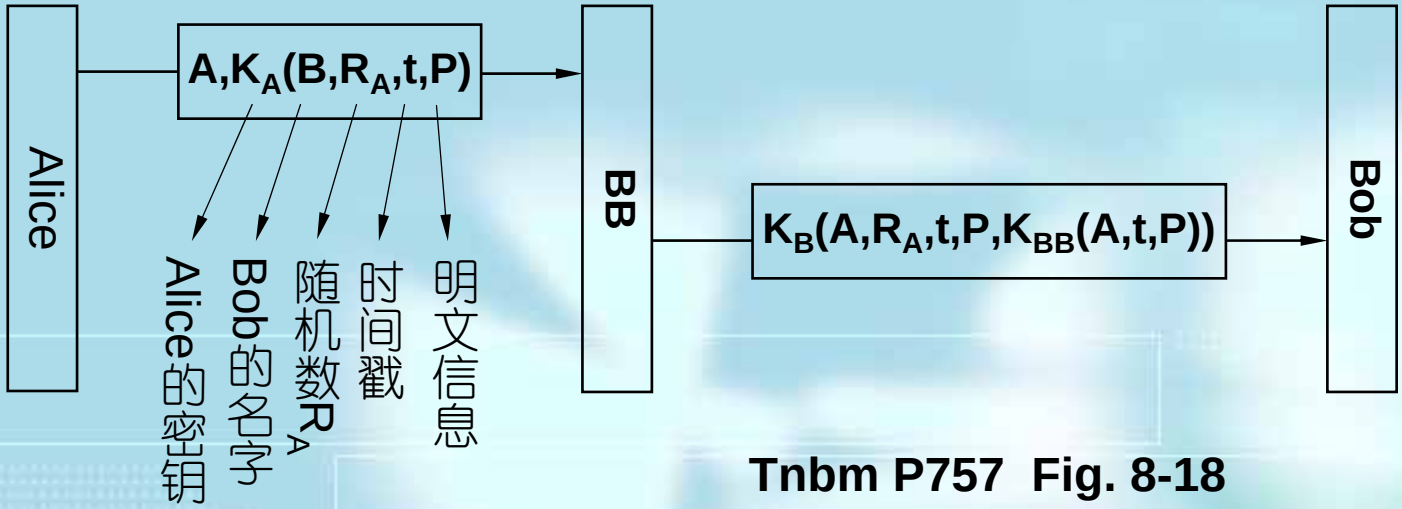
- 密码系统
- 对称密钥算法
- 公钥算法
- 数字签名
- 公钥管理
- 通信安全
- 认证协议
- **E-mail的安全**
- **Web安全**

数字签名

数字签名的目的

- 接收方能够验证发送方所宣称的身份
- 发送方以后不能否认报文是他发的
- 接收方不能伪造该报文

采用对称密钥的数字签名

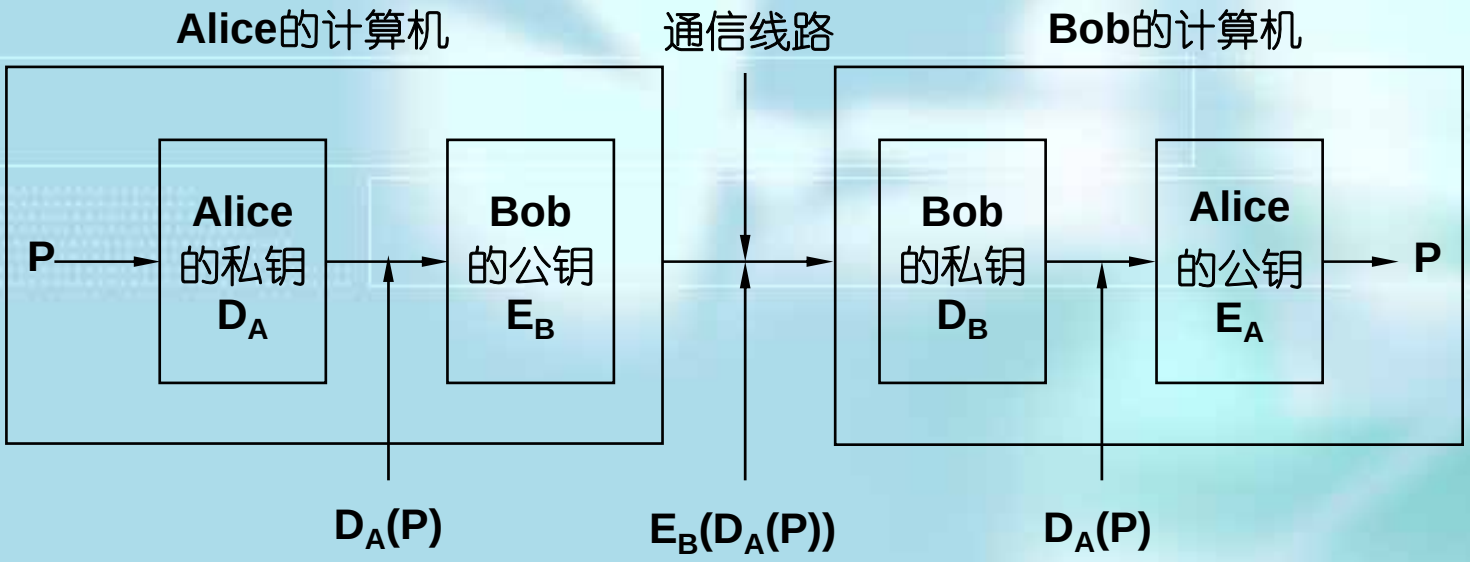


Tnbm P757 Fig. 8-18

- 一个公认的信任机构**BB**，负责给每个人分配密码
- 传输时，也必须通过该信任机构，如**A**发一消息给**B**，他必须先用自己的密钥加密后发给信任机构**BB**，信任机构**BB**解密，然后重新用**B**的密钥加密后发给**B**

采用公开密钥的数字签名

- 秘密密钥加密的问题：需要有公认的信任机构，但事实上很难找到这样的机构



Tnbm P58 Fig. 8-19 公开密钥的数字签名

报文摘要

- 提供一种不需要对完整的信息进行加密就能达到认证的目的的方法
- 报文摘要(MD)是基于一个单向的hash函数，从明文中取出任意长的部分，从中计算出一个定长的bit串
- 报文摘要的特性
  - 给定P，很容易就能计算出MD(P)
  - 给定MD(P)，不可能推算出P
  - 给定P，不可能发现一个P'并使得MD(P) = MD(P')
  - 当输入改变时，甚至改变一个Bit，都将产生不同的输出

报文摘要

- 利用报文摘要进行签名
  - 用对称密钥
  - 用非对称密钥
- 签名后的摘要随明文一起发送
- 最常用的报文摘要是**MD5**和**SHA-1**

本章将讨论：

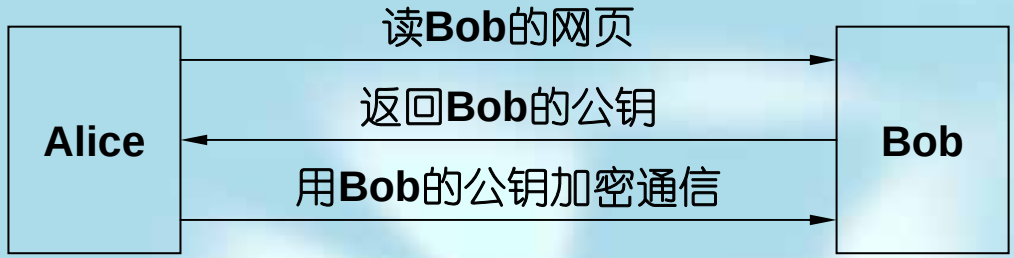
- 密码系统
- 对称密钥算法
- 公钥算法
- 数字签名
- 公钥管理
- 通信安全
- 认证协议
- E-mail的安全
- Web安全

公钥管理

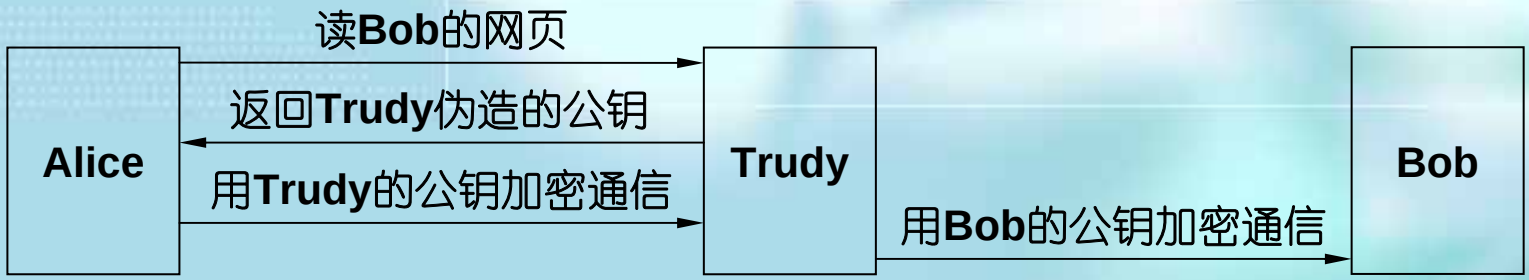
本节讨论两个互不相识的人如何通过公钥机制来通信？如何能得到对方的公钥？

- 公钥获取中的安全问题
- 证书
- **X.509**
- **PKI**（公钥基础设施）

公钥获取中的安全问题



正常情况下的通信



有非法者的情况下的通信

Tnbm P765 Fig. 8-23 Trudy 破坏公开密钥加密的一种方法

公钥管理

本节讨论两个互不相识的人如何通过公钥机制来通信？如何能得到对方的公钥？

- 公钥获取中的安全问题
- 证书
- **X.509**
- **PKI**（公钥基础设施）

证书

- 设置一个机构**CA**（**Certification Authority**）证明某些公钥是属于某个人或某个机构，这个证明称作为证书
- 证书用**SHA-1**做摘要，该摘要用**CA**的私钥加密
- 证书的拥有者可将证书放在网上，供希望与他通信的人下载
- 证书可解决伪造者的问题

- 伪造者用自己的证书替换**Bob**的证书：由于证书中有持有者姓名，**Alice**马上就可发现有人伪造
- 伪造者用自己的公钥替换**Bob**的证书中的公钥：由于证书是作过摘要，并用**CA**的私钥加密，通过摘要可检查出证书被修改

公钥管理

本节讨论两个互不相识的人如何通过公钥机制来通信？如何能得到对方的公钥？

- 公钥获取中的安全问题
- 证书
- **X.509**
- **PKI**（公钥基础设施）

X.509

- X.509是ITU制定的证书标准，它包括以下字段

字段	意义
版本	X.509的版本号
系列号	这个编号加上CA的名字唯一确定这个证书
签名算法	用来签署这个证书的算法
发布者	CA的X.509名字
有效期	证书的有效期
持有者姓名	持有者姓名
公钥	持有者的公钥和所用算法的编号
发布者的ID	唯一确定发布者的编号
持有者ID	唯一确定持有者的编号
扩展	已定义的许多扩展
签名	证书的签名（用CA的私钥签名）

Tnbm P768 Fig. 8-25 X509的证书标准

公钥管理

本节讨论两个互不相识的人如何通过公钥机制来通信？如何能得到对方的公钥？

- 公钥获取中的安全问题
- 证书
- **X.509**
- **PKI**（公钥基础设施）

PKI（公钥基础设施）

- 为了解决认证过程中的一些问题，如单个认证中心负担太重，多个认证中心容易引起证书的泄漏，认证中心应该由哪个机构来运作等问题
- **PKI**的组成：由用户、**CA**、证书和目录组成，**PKI**提供如何将这些机构组织起来，如何定义各种文件和协议的标准
- 最简单的结构是层次结构，有根、**RA**（区域机构）和**CA**组成

本章将讨论：

- 密码系统
- 对称密钥算法
- 公钥算法
- 数字签名
- 公钥管理
- 通信安全
- 认证协议
- E-mail的安全
- Web安全

通信安全

- IPSec
- 防火墙
- VPN
- 无线安全

IPsec

- 提供为多种服务、多种算法和多种粒度的加密框架
- 多种服务指的是安全、完整和抗重播，所有的这些都是基于对称加密
- 多种算法指的是算法是独立的，即使某些算法将来被攻破，**IPsec**的框架还是保持不变
- 多种粒度指的是可以保护单个的**TCP**流，也可以保护一对主机间的所有流量或以对安全路由器间的所有流量

安全联盟SA（Security Association）

- 虽然IPsec是工作在IP层，但它却是面向连接的，一个连接被称为一个安全联盟SA
- SA是单向的，如要进行双向通信，必须要两个SA
- 每个SA有一个与之相关的安全标识符，安全标识符被放入该安全通道中的每个数据包中，用来检查密钥和一些相关的信息

IPsec的组成

- 用来携带安全标识、完整性控制和其他数据的两个新头部
- 安全联盟和密钥管理协议**ISAKMP**用来处理创建密钥的工作

IPsec的工作方式

- 原始的IP包

IP头	TCP头	数据

- 传输模式

IP头	IPsec头	TCP头	数据

- 通道模式

IP头	IPsec头	IP头	TCP头	数据

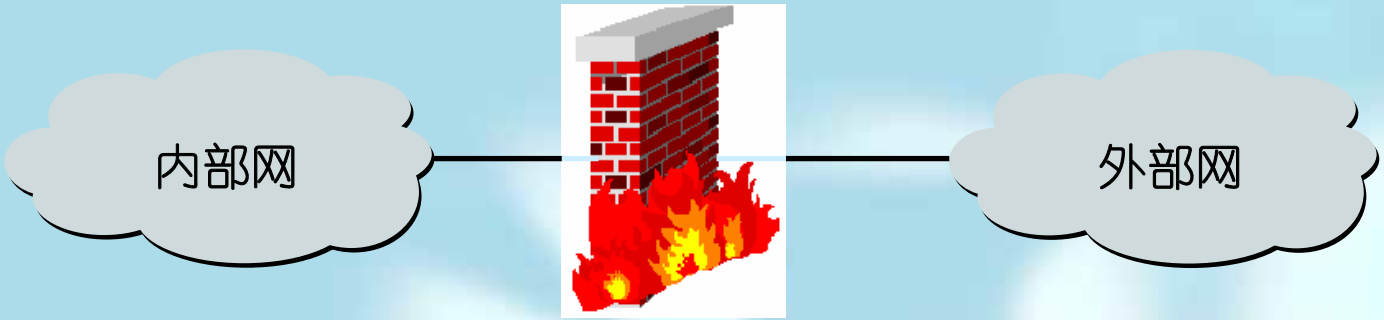
IPsec的头部

- 封装安全载荷**ESP**（**Encapsulating Security Payload**）：属于**IPsec**的一种协议，可用于确保**IP**数据包的机密性（未被别人看过）、数据的完整性以及对数据源的身份验证。此外，它也要负责对重播攻击的抵抗
- 验证头**AH**：用于为**IP** 提供数据完整性、数据原始身份验证和一些可选的、有限的抗重播服务，但不提供加密功能

通信安全

- IPSec
- 防火墙
- VPN
- 无线安全

防火墙

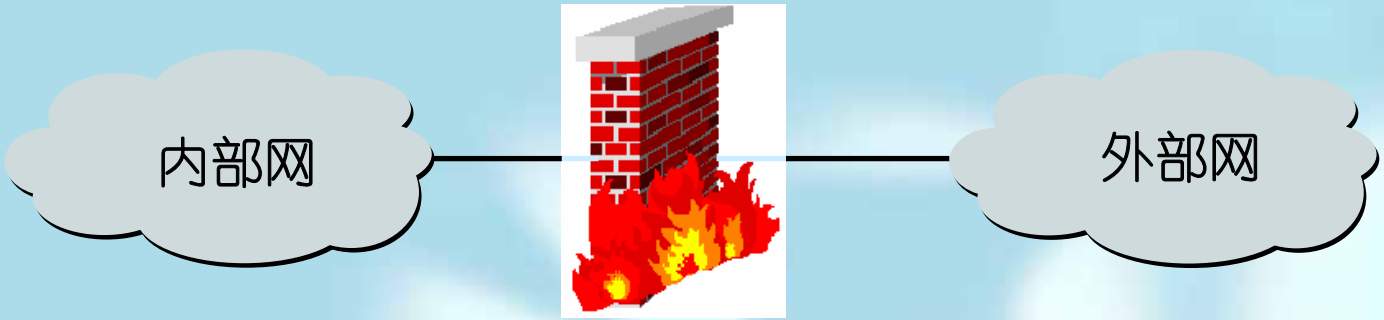


- 防火墙的作用
- 基于协议层的防火墙分类
- 网络层防火墙
- 应用层防火墙

防火墙的作用

- 保障内部网的安全，不受攻击
- 监视、记录进出内部网的信息，包括流量统计，设置访问控制表等
- 可以设置**NAT(Network Address Translate)**网络地址转换器，用于节约**IP**地址，使大量用户使用少量**IP**地址，让用户仅在出子网时使用正式的**IP**地址，否则使用内部的**IP**地址
- 可采用加密技术对信息进行加密处理

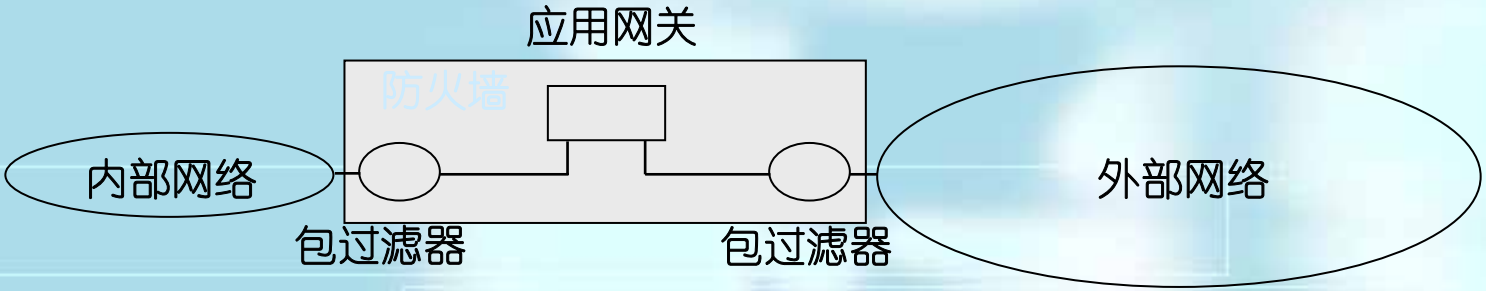
防火墙



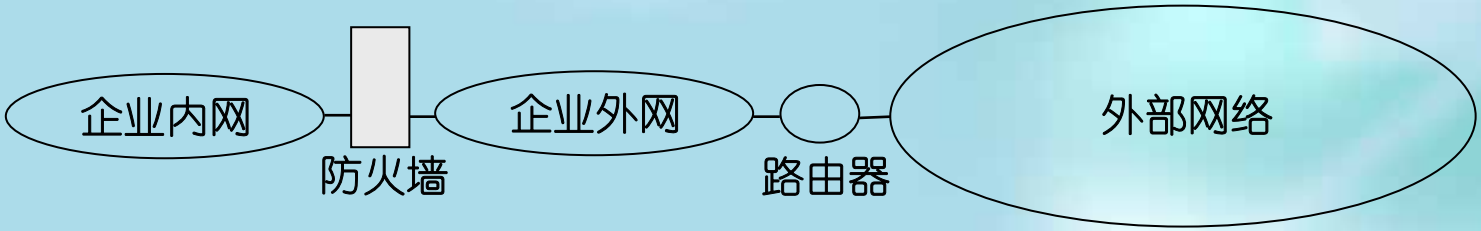
- 防火墙的作用
- 基于协议层的防火墙分类
- 网络层防火墙
- 应用层防火墙

基于协议层的防火墙分类

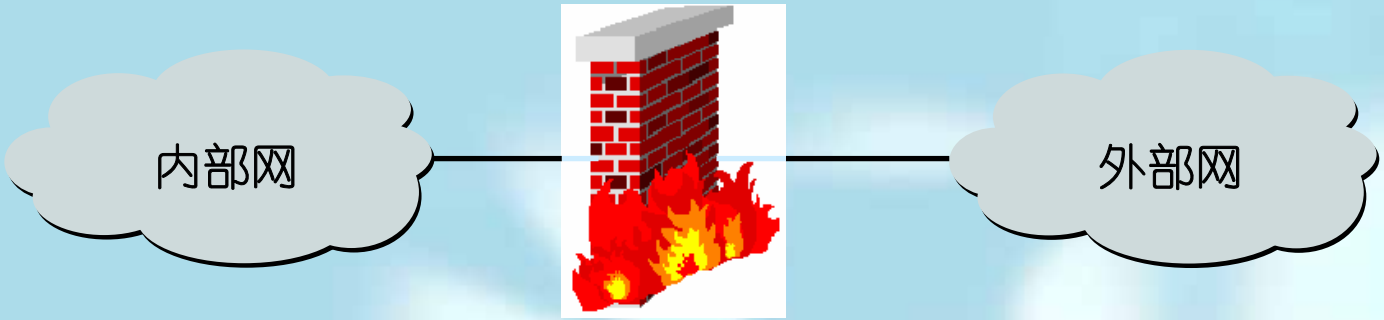
- 包过滤器：网络层
- 应用网关：应用层



在实际应用中，基于网络层的防火墙的安装位置



防火墙



- 防火墙的作用
- 基于协议层的防火墙分类
- 网络层防火墙
- 应用层防火墙

网络层防火墙

- 检查的项目

  源IP地址

  目的IP地址

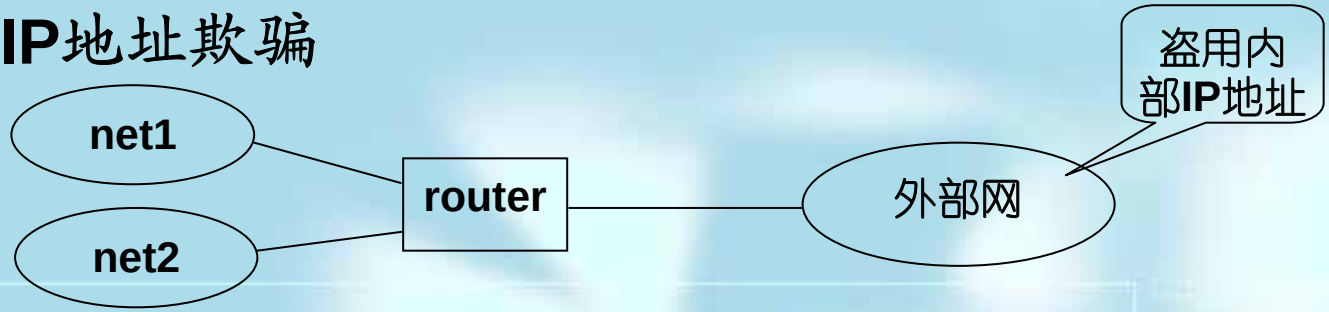
  TCP/IP协议及其源、目的端口号(port number)

- 访问控制表（+表示无限制）

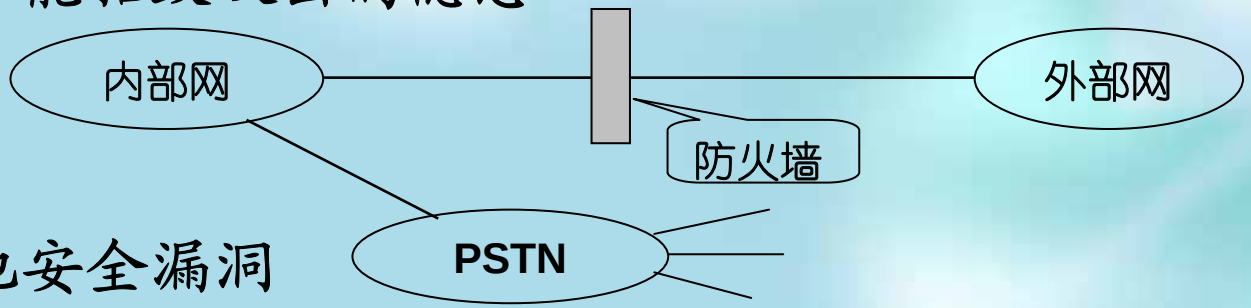
action	src	port	des	port	flag	comments
allow	+	>1023	202.120.10.1	23		telnet
allow	+	>1023	202.120.10.2	25		SMTP
allow	212.5.32.6	>1023	202.120.10.3	119		NNTP
allow	+	+	+	+	ACK	回答响应
block	hackers address	+	+	+		

网络层防火墙（续）

- 特殊情况的漏洞
  - IP地址欺骗

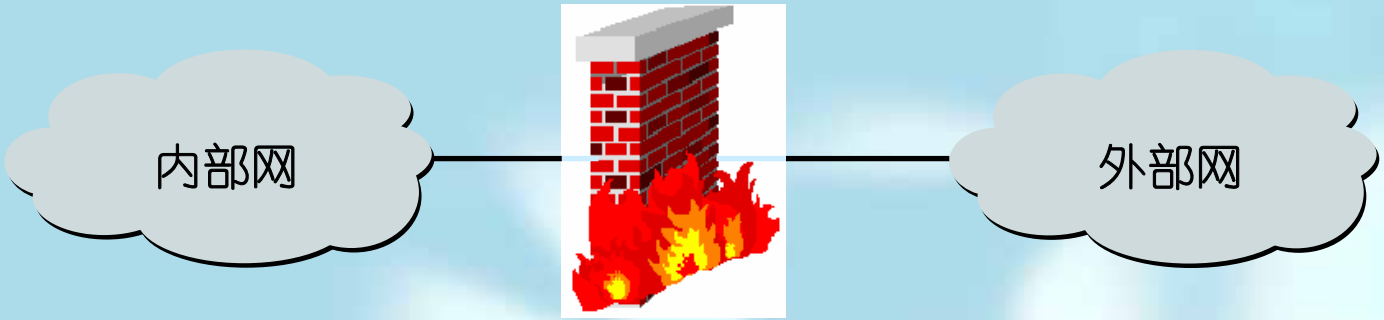


  - IP碎片处理：没有包含原报文头的信息的碎片被拒收，因TCP头部中的端口号不在碎片中，但这是可能招致攻击的隐患



  - 其他安全漏洞

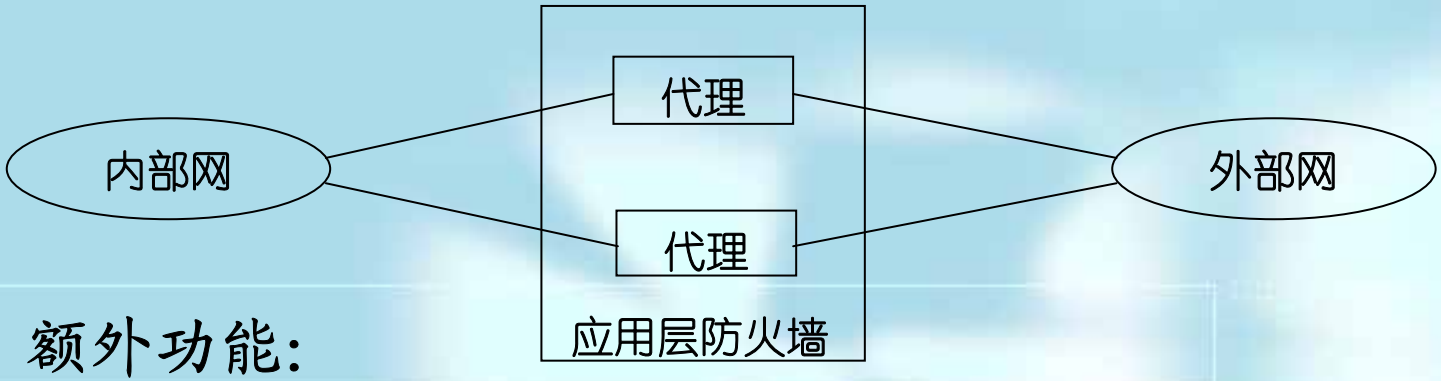
防火墙



- 防火墙的作用
- 基于协议层的防火墙分类
- 网络层防火墙
- 应用层防火墙

应用层防火墙

- 采用代理网关，外部网委托代理执行相应的操作



额外功能：

- 代理可控制一些服务的子功能，如**FTP**，可设置服务器只提供**get**不提供**put**
- 流量、计费等功能
- 检查传输信息本身，如**mail**

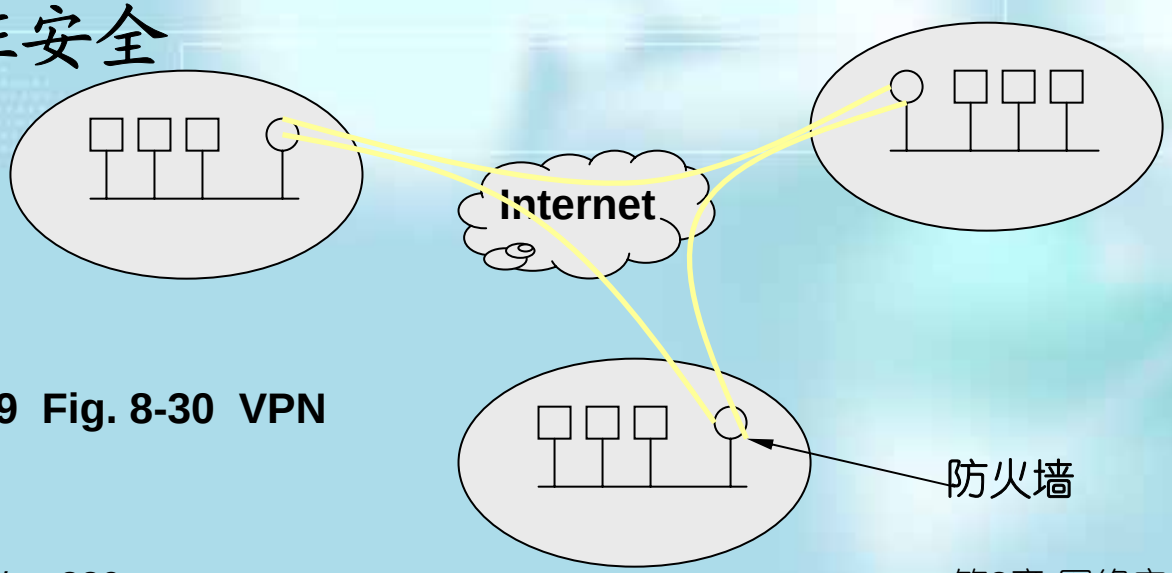
对不同的应用，应建立不同的应用网关，开销较大，所以通常仅开放几个常用的应用

通信安全

- IPSec
- 防火墙
- VPN
- 无线安全

VPN

- 在公共网络中建立专用的数据通信网的技术，以取代原来的专线
- **VPN**可建在**ATM**或帧中继上，但目前一般指的是建立在**Internet**上，通过防火墙和隧道技术保证安全



Tnbm P779 Fig. 8-30 VPN

通信安全

- IPSec
- 防火墙
- VPN
- 无线安全

无线安全

- **802.11**的安全：定义了一个数据链路层的安全协议**WEP**（**Wired Equivalent Privacy**）
- 蓝牙的安全:
- **WAP**:

本章将讨论：

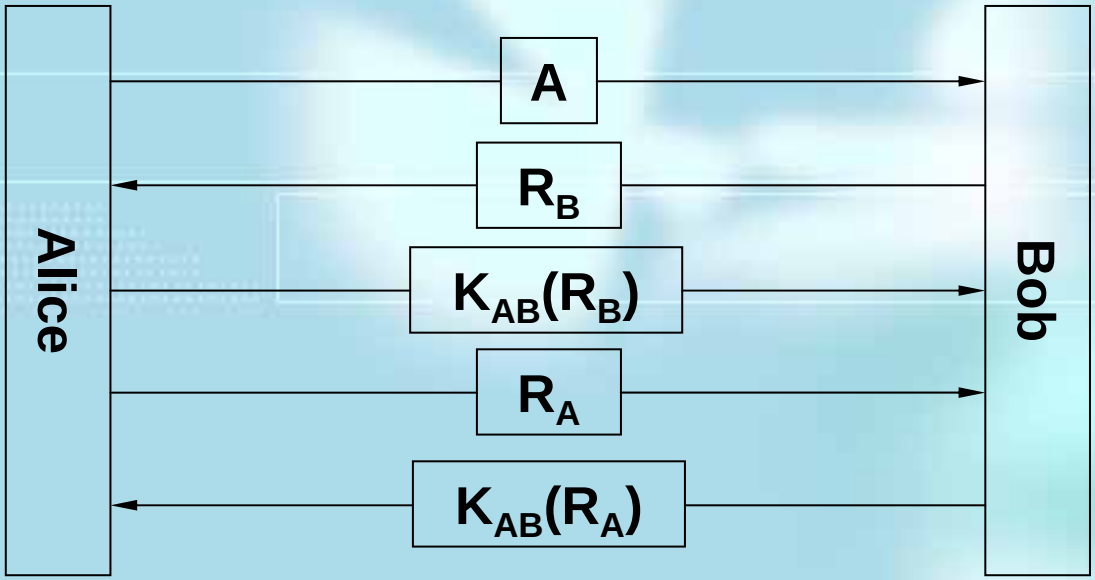
- 密码系统
- 对称密钥算法
- 公钥算法
- 数字签名
- 公钥管理
- 通信安全
- 认证协议
- **E-mail的安全**
- **Web安全**

认证协议

- 如何验证通信的对方是约定者而不是入侵者
- 两种认证方法
- 基于公钥：用**PKI**的机制
- 基于对称密钥

基于共享密钥的认证

- 通信双方事先已约定一个密钥，这种情况下的认证可用**challenge-response**协议



Tnbm P787 Fig. 8-32 使用challenge-response协议认证的两种方法

创建一个共享密钥—Diffie-Hellman密钥交换

- 在一个不保密的、不受信任的通信信道上（比如**Internet**），在交换的双方之间，建立起一个安全的共享秘密的会话
- **Diffie-Hellman**交换过程中涉及到的所有参与者首先都必须隶属于一个组，这个组定义了要使用哪个质数**p**，以及底数**g**

Diffie-Hellman密钥交换过程

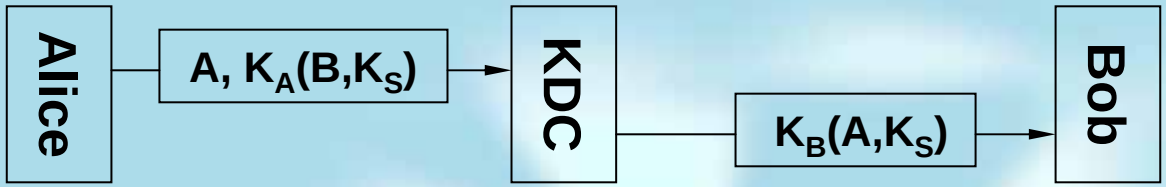
- 在每一端（如Alice和Bob进行通信）的第一过程中，需要选择一个大的随机的私人数字（例如512 bit），并在组内进行乘幂运算，产生一个公共值，例如：Alice选择了a，并计算$A = g^a \bmod p$；Bob选择了b，并计算$B = g^b \bmod p$
- 他们开始交换自己的公共密钥，Alice将A给Bob，而 Bob将B给Alice，他们分别再次执行乘幂运算，由于 $B^a \bmod p = A^b \bmod p = g^{ab} \bmod p = w$, 所以双方的计算结果相同，这个结果值 w 就是Alice和Bob的公钥

Diffie-Hellman交换过程举例

- Alice选定大素数n = 47，g = 3和自己的密钥x = 8；并计算：$g^x = 3^8$，$3^8 \bmod 47 = 28$；最后将(47, 3, 28)发送给Bob；
- Bob选定自己的密钥y = 10，并收到Alice发来的 (47, 3, 28)，知道Alice选用的大素数n = 47，g = 3，$g^x \bmod n = 28$
- Bob计算$(g^x \bmod n)^y = 28^{10}$，$28^{10} \bmod 47 = 4$，即为他俩的公钥，然后Bob计算$g^y = 3^{10}$，并把$3^{10} \bmod 47 = 17$发给Alice
- Alice收到Bob发来的17，知道Bob的$g^y \bmod n = 17$，经计算$(g^y \bmod n)^8 = (17)^8 \bmod 47 = 4$，即为他俩的公钥

用密钥分发中心KDC的认证

- 每个用户与KDC有一个共享密钥



Tnbm P793 Fig. 8-39

- Alice选择一个session key：K_S
- Alice告诉KDC要与Bob用K_S加密通信，这条消息用Alice与KDC共享的密钥K_A加密
- KDC对其解密，重构一条包含Alice的标识 A 以及session key K_S的报文发给Bob，这条消息用Bob与KDC的共享密钥K_B加密
- Bob收到这条消息，得知Alice要与它用session key K_S通信

本章将讨论：

- 密码系统
- 对称密钥算法
- 公钥算法
- 数字签名
- 公钥管理
- 通信安全
- 认证协议
- **E-mail的安全**
- **Web安全**

E-mail的安全

- PGP：Internet上的一个免费软件包，提供加密、认证、数字签名和压缩
- PEM：Internet的标准，定义在RFC1421和RFC1424中，使用者较少
- S/MIME：定义在RFC2632和RFC2643中

本章将讨论：

- 密码系统
- 对称密钥算法
- 公钥算法
- 数字签名
- 公钥管理
- 通信安全
- 认证协议
- **E-mail的安全**
- **Web安全**

WEB安全—安全命名

- **MAN-IN-MIDDLE**攻击：当**Alice**要访问**Bob**的网页时，他的请求被**Trudy**截取，并伪造网页返回给**Alice**
- **DNS**欺骗：修改域名服务器中的域名-地址对照表，该修改可以通过伪造**DNS**应答报文来实现，当**Alice**要访问**Bob**的网页时，域名服务器将返回一个伪造的地址

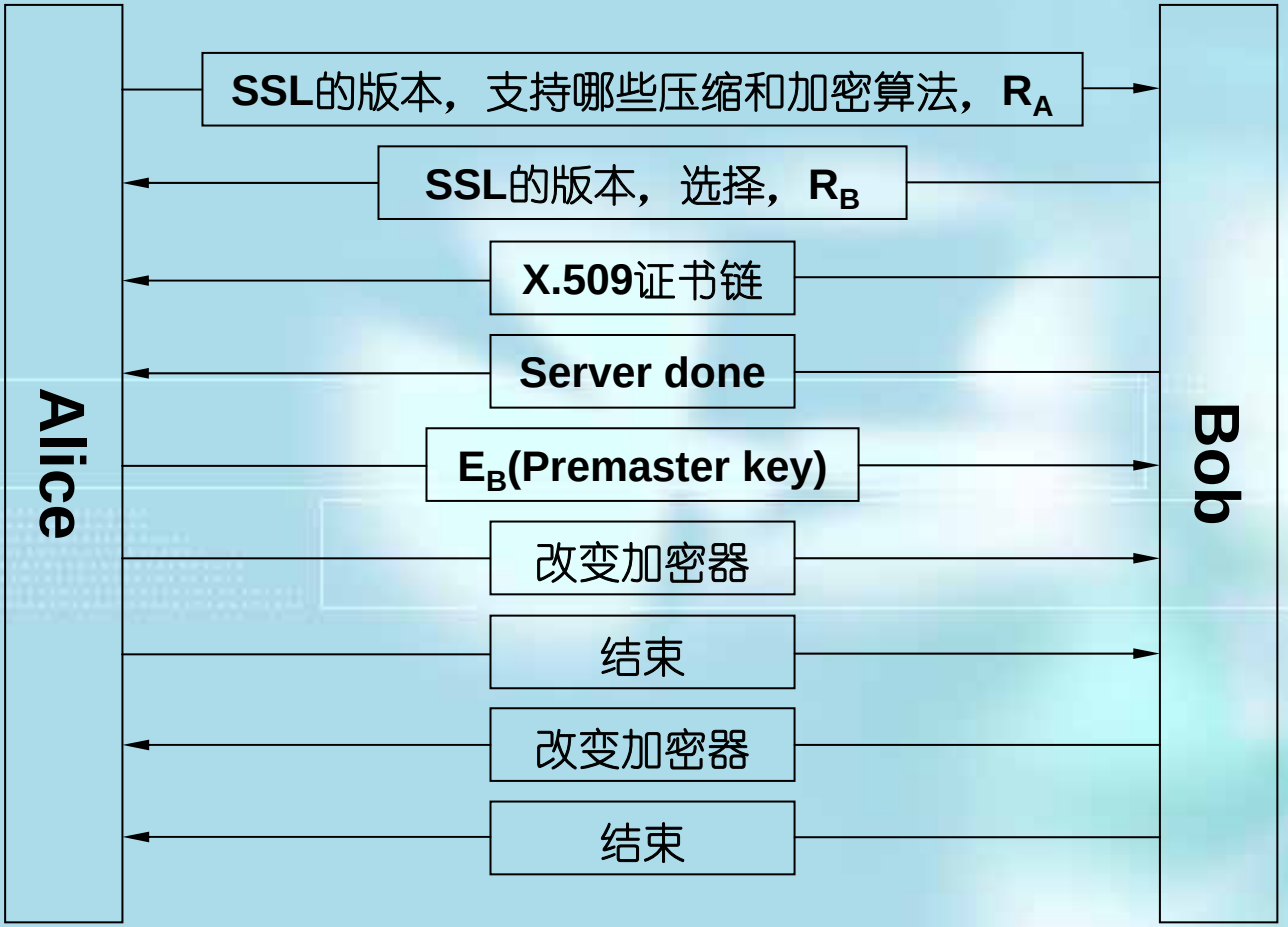
WEB安全—安全命名

- 安全DNS：每个DNS域都有一个公钥/私钥对，所有有DNS服务器发出的信息均用私钥签名，接收者可用公钥认证
- 自认证域名：在URL中域名和文件名之间插入一个加密的服务器名的散列值，当用户使用该URL时，浏览器会向对应的网站请求公钥，收到公钥后，浏览器运行相应的HASH算法，如结果相同，浏览器认为该站点是真正要访问的站点，则发送该URL

WEB安全—安全连接

- 安全连接通常使用安全套接字协议**SSL**，它位于应用层与传输层之间
- **SSL**在两个套接字之间建立一条安全通道，包括：
  - 客户/服务器之间参数协商
  - 客户/服务器之间互相认证
  - 加密通信
  - 数据完整性保护
- **SSL**包括两个子协议：建立安全通道和使用安全通道

WEB安全—安全连接

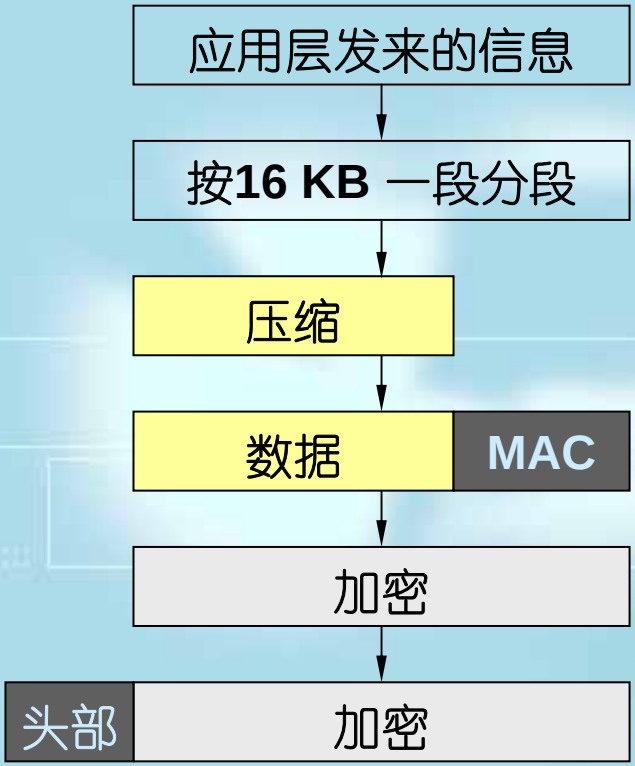


Tnbm P815 Fig. 8-51 连接建立过程

WEB安全—安全连接

- Alice给Bob发送自己的SSL版本，支持哪些加密和压缩算法以及一个时间值R_A
- Bob在Alice支持的算法中做出选择，将此选择和一个时间值R_B
- Bob给Alice发送公钥的认证链
- Bob发送Server done表示结束
- Alice选择一个384位的Premaster key，并发送给Bob，从Premaster key、R_A和R_B可计算Session key
- Alice通知Bob改变加密器
- Alice通知Bob发送结束
- Bob给Alice发送应答

WEB安全—安全连接



SSL传送过程

WEB安全—移动代码的安全

- **Java applet**的安全：由于用解释方法执行，可限制某些系统调用的执行
- **Active X**：是一段二进制代码，无法逐条检查，只能选择执行或不执行，**MS**的方法是用数字签名，如果是认可的代码则执行，否则不执行
- **Java Script**：无通用的解决方法
- 病毒：无解决方法

第8章 习题

Tnbm P829

#14(七 #8)

#31(七 #9)