网络信息安全复习总结

——YCIrving

大三课业负担不是很紧，所以有时间来认真复习一下每一科。看着书店各种大神写的复习攻略，自己也想试着来写一份，希望能够对自己甚至对学弟学妹们有所帮助。

首先必须明确一下考试的题型，15 年软件工程网络信息安全的命题人说这次考试的题型共五类，即选择、简答、计算、证明、分析（具体的分值分配没有说），下面根据本书的一些知识点再结合题型，来说一下具体的复习方法。

选择题(五道)考的知识点应该比较小，也比较难掌控，只有认真把书看一遍并且多看看试题才能做到胸有成竹。下面是我自己的一些知识点总结：

1. 安全机制指用来**检测、阻止**攻击或者从攻击状态里**恢复**的**过程**(或包含这种过程的设备) ，**最重要的安全机制之一就是密码编码机制**
2. 欲达理论安全，加密密钥长度必须**大于等于**明文长度，密钥只用**一次**，用完即丢，即**一次一密**，不实用。
3. 密码体制：加密系统采用的基本工作方式称为密码体制。密码体制的基本要素是**密码算法**和**密钥**。密码算法是一些公式、法则或程序；密钥是密码算法中的控制参数。(**明文密文不算基本要素，**注意跟书 22 页上对称密码模型 5 个基本成分的区别)

4. 序列密码(流密码)和分组密码

• 序列密码：如果密文不仅与最初给定的算法和密钥有关，同时也与**明文位置有关**(是所处位置的函数)，则称为序列密码体制。加密以明文**比特**为单位，以伪随机序列与明文序列模 2 加后，作为密文序列。

• 分组密码：如果经过加密所得到的密文仅与给定的密码算法和密钥有关，与被处理的明文数据在整个明文中的**位置无关**，则称为分组密码体制。通常以大于等于 64 位的数据块为单位，加密得相同长度的密文。

(即流密码在加密时即使明文相同，但如果位置不同，加密的结果也不同，因为明文加密时是使用密钥流的不同部分加密的；而分组加密时，如果分组的明文相同，即使位置不同，加密结果也相同，因为对于每一分组，加密都是用相同的密钥加密的)

5. 确定型和概率型密码体制，单向函数型和双向变换型密码体制

确定型：明文相同，加密后密文一定相同；

概率型：明文相同，加密后密文不一定相同；

单向函数型：不需要解密，如哈希函数；

双向变换型：可以进行加解密之间的变换。

6. 现代密码学基本原则

设计加密系统时，总是假定密码**算法**是可以**公开**的，需要**保密**的是**密钥**。一个密码系统的**安全性不在算法的保密，而在于密钥。**

7. 对称加密系统的五个组成部分：明文、加密算法、密钥、密文、解密算法
8. 所有加密算法都基于两个原理：**代替和置换。**代替是将明文中的每个元素映射成另一个元素，置换是将明文元素重新排列。

9. 密码攻击的两种方式：密码分析学攻击(差分分析和线性分析)和穷举攻击。
10. 另一种密码攻击的分类：

唯密文攻击、已知明文攻击、选择明文攻击、选择密文攻击、选择文本攻击(见书 25 页)，一般的，加密算法起码要能经受得住**已知明文攻击**。

11. 凯撒密码的密钥数为 **25 个**，它是一种简单的**单表代替密码**。
12. 一次一密的两个限制：产生大规模随机密钥有实际困难、密钥的分配和保护无法保证。
13. 隐写术不是加密技术。
14. DES 的**分组**和**密钥**分别是 **64 位**和 **56 位**，整个加密包含 **16 轮迭代**。
15. S 盒：**8 个**，将 **6 位**数据映射成 **4 位**数据(即 6 位输入，4 位输出) 。
16. 扩展欧几里得算法可以用来计算**乘法逆元。**
17. 多次加密的最简单形式是进行两次加密，每次使用不同的密钥，但存在**中间相遇攻击。**
18. 使用两个密钥进行三次加密：**E-D-E**，即 $\mathbf{C=E_{K1}[D_{K2}[E_{K1}[P]]}$；三个密钥加密，$\mathbf{C = E_{K3}[D_{K2}[E_{K1}[P]]]}$。
19. 数论的核心是**素数。**
20. 应用**最广泛**的公钥密码体制是 **RSA**，破解 RSA 的困难，是**基于分解大合数的素因子**的困难。
21. 公钥密码体制中，**穷举消息攻击**是第三种攻击形式，攻击者用公钥对所有可能的消息加密，并与传送的密文匹配，从而解密任何消息；抵抗的方法是在要发送的消息后附加随机数(即对传送的明文进行穷举攻击)。
22. X.509 有三种可选的认证过程：One-Way、Two-Way、Three-Way(三种方法均使用公钥签名)
23. PKI 的操作模型

PKI 中有**两种管理实体**：证书管理中心 CA 和注册中心 RA

CA：能够发布和撤销证书，维护证书的生存周期

RA：负责处理用户注册请求，在验证请求有效性后代用户向 CA 提交。

PKI 中有**两种端实体：持证者(Holder)和验证者(Verifier)**

持证者是证书拥有者，是证书所声明事实的主体；验证者通常是授权方，确认证书持有者的证书有效后再授予对方相应权力。

不同实体之间通过 PKI 操作完成证书的请求、确认、发布、撤销、更新、获取等过程。**PKI 的操作主要分成两大类：存取操作和管理操作。**

简答题(一道)问的东西就需要自己多加理解了，主要是一些基本概念性的知识点，必要时也需要加强一下记忆，不需要预备知识，可能考到的知识点如下（有关对具体协议等步骤的考察我放在了分析题中，这里不涉及）：

1. 最根本的，密码学就学了三块，对称密码、非对称密码和其他（包括哈希函数、MAC、认证和数字签名等），所以前两者的区别一定要知道：

•对称密码体制：加密密钥和解密密钥相同，或者一个密钥可以从另一个导出，能加密就能解密，加密能力和解密能力是结合在一起的，密钥更换、传递和交换需要可靠信道，密钥分发困难，开放性差。如有 N 用户，则需要 C=N (N-1)/2 个密钥，n=1000 时，C(1000, 2)≈500000, 密钥管理困难，无法满足不相识的人之间通信的保密要求，不能实现数字签名

• **非对称密码体制：加密密钥和解密密钥不相同**，从一个密钥导出另一个密钥是计算上不可行的，加密能力和解密能力是分开的，密钥分发简单，开放性好。需要保存的密钥量大大减少，N 个用户只需要 N 个密钥，可满足不相识的人之间保密通信，可以实现数字签名。

说明：加解密密钥是否形同是二者最根本的区别。

2. 安全攻击分为被动攻击和主动攻击。

• **被动攻击：**通信和信息不受影响，用户感觉不到攻击存在。

包括：消息内容泄露、流量分析(判断通信性质)

• **主动攻击：**攻击者破坏通信过程

包括：伪装、重放、消息修改、拒绝服务

说明：安全攻击本身的定义不会考，但是它的两种分类需要掌握。

3. **安全服务**包括**认证、访问控制、数据保密性、数据完整性、不可否认性**以及可用性。(X.800 将安全服务分为 5 类共 14 个特定服务)

另外，知道攻击类型对应哪种服务也很可能在选择题中出现，比如伪装攻击对应认证服务、信息内容泄露和流量分析对应数据保密性、修改重播拒绝服务对应数据完整性、**拒绝服务对应可用性服务。**

4. 理论安全和实际安全

• **理论安全(无条件安全)：**攻击者无论截获多少密文，都无法得到足够的信息来唯一地决定明文。

• **实际安全(计算上安全)：**在有限的资源范围内，攻击者都不能通过系统的分析方法来破解系统，则称这个系统是计算上安全的或破译这个系统是计算上不可行。

5. 密码编码系统根据以下三个独立方面进行分类：

(1)、转换明文为密文的运算类型；(代替和置换)

(2)、所用的密钥数；(相同：对称；不同：非对称、公钥)

(3)、处理明文的方法。(分组密码和流密码)

6. Feistel 结构：由许多相同的轮函数组成。每一轮里，对输入数据的一半进行**代换**，接着用一个**置换**来交换数据的两个部分，**扩展初始的密钥**使得每一轮使用**不同的子密钥**。

7. 扩散：明文统计结构扩散消失到大批密文统计特性中，使**明文和密文**之间统计关系尽量复杂(即明文一比特变化可能引起密文较多比特变化)；
混淆：使**密文和加密密钥**之间的关系尽量复杂。

8. 分组密码的五种工作模式：

• 电子密码本模式（ECB）：明文分成 64 的分组进行加密，必要时填充，每个分组用同一密钥加密，同样明文分组加密得相同密文。

• 密文分组链接模式（CBC）：加密输入是当前明文分组和前一密文分组的异或，形成一条链，使用相同的密钥，这样每个明文分组的加密函数输入与明文分组之间不再有固定的关系

• 密文反馈模式（CFB）：是一种将 DES 转化成流密码的技术。加密函数高端 j 位与明文 P1 的第一单元异或，产生 j 位密文 C1 进入移位寄存器低端，继续加密，与 P2 输入异或，如此重复直到所有明文单元都完成加密。

• 输出反馈模式（OFB）：结构上类似 CFB，但是 OFB 中加密函数输出被反馈回移位寄存器。

• 计数器模式（CRT）：每一个明文分组都必须使用一个不同的密钥和计

数器值，决不要重复使用

上面介绍的概念比较抽象，插图的话有比较浪费页面，所以建议大家对照着书，把每一种工作模式的流程都理解了，下面我给出一个表，总结一下它们各自的特点：

名称	支持流密码	是否需要填充	比特差错传播	明文相同密文是否相同	其他特点
ECB	否	是	否	是	**适合数据较少的情况； 明文相同，密文也相同。**
CBC	否	是	是	否	**明文一点变化能引起所有密文变化； 可通过改变 IV 来实现对明文第一个分组特定位的修改。**
CFB	是	否	是	否	**密钥先与明文异或，再将结果送入移位寄存器。**
OFB	是	否	否	否	**将密钥先送入寄存器，再与明文异或；由于没有差错传播，所以更易受报文流篡改攻击。**
CTR	否	否	否	否	**密钥不重复使用，可同时加密，适用于高速网络。**

9. 公钥密码体制的应用

加密/解密：发送方用接收方的公钥对消息加密

数字签名：发送方用其私钥对消息签名，可以对整体消息签名或对消息的摘要签名

密钥交换：通信双方交换会话密钥

10. 公钥密码体制中的密钥分配

公钥密码的主要作用之一就是解决**密钥分配**问题(因为如果用来加密信息效率不高，所以大多数信息的加密还是对称加密)，密钥分配问题主要集中在以下两个方面：公钥的分配、对称密码体制的密钥分配，**针对前者**，有以下几种公钥分配方法：

名称	内容	存在问题
公开发布	用户将他的公钥发送给另一通信方，或者广播给通信各方。	最大问题在于任何人都可以伪造这种公钥的发布。
公开可访问的目录	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。	一旦攻击者获得目录管理员私钥，则可传递伪造的公钥，可以假冒任何通信方以窃取消息，或者修改已有的记录。
公钥授权	具体分析见**分析题。**	也有被篡改的风险；公钥授权中心必须实时在线，成为一个网络交换的**瓶颈**。
公钥证书	涉及服务为 X.509，”只看不改”,具体分析见**分**	没错，它就是最好的！

	析题。	

针对后者主要是应用 Diffie-Hellman 密钥交换协议(协议存在的中间人攻击也会在分析题中讨论)。

11. **消息认证**是用来验证**消息完整性**的一种机制或服务。消息认证确保收到的数据确实**和发送时的一样**(即没有修改、插入、删除或重放)，且发送方声称的身份是真实有效的，同时也要求信息源不可否认。**对称密码**和**私钥**都可以提供认证服务，用于消息认证的最常见的密码技术是**消息认证码**(MAC)和**安全散列(hash)函数**。其中 MAC 是一种**需要使用秘密密钥**的算法，而 Hash 则不需要密钥，因此它必须以某种方式和秘密密钥**捆绑起来使用**。

12. 消息认证中哈希函数的要求：

(1)H 可以应用于任意大小的数据块 **输入任意**、

(2)H 产生固定长度的输出 **输出固定**、

(3)对任意给定的明文 x，计算 H(x)容易，可由硬件或软件实现 **计算容易**、

(4)对任意给定的散列码 h，找到满足 H(x)=h 的 x，在计算上不可行，**单向性**、

(5)对任何给定的分组 x，找到满足 y≠x 且 H(x)=H(y)的 y，在计算上不可行，**抗弱碰撞性**、

(6)找到任何满足 H(x)=H(y)的偶对(x, y)，在计算上不可行，**抗强碰撞性**

条件 6 主要用于防范所谓的**生日攻击法**，满足条件 1-5 的称为**弱哈希函数**，满足 1-6 的称为**强哈希函数**，应用在**数字签名**上的必须是**强哈希函数**。

13.数字签名：**数字签名是一种认证机制**，它使得消息的产生者可以添加一个起签名作用的码字。通过计算消息的散列值并用产生者的私钥加密散列值来生成签名。签名保证了消息的来源和完整性。

与消息认证的区别：消息认证是使消息接收方验证消息发送者发送的内容有无被修改过，对防止第三者破坏足够，**但收发双方有利害冲突时就无法解决纷争，需要更严格的手段，即数字签名**。

签名的两种方式：对消息整体签名，对消息摘要签名；

签名还分确定性(明文密文一一对应)和概率性(一个明文可对应多个签名)；

计算题(两道)考的东西无非就是数论里面的两个定理以及后来非对称加密的几个算法，考前多用笔亲自练习，稍加理解即可，但是切忌眼高手低：

预备知识点：

• 整除的读法和表示：

a|b 读作 a 整除 b，含义是 a 是 b 的一个因子，即 b/a=0；（可以用这个表示跟分数正好相反来记忆）。

• 欧几里得算法求最大公约数的过程：

Eg:gcd(18,12)=gcd(12,6)=gcd(6,0)=6;

其实如果用代码来记，更能让计算过程机械化：

```
Euclid(a,b)//起始时一般让 a 大于 b，不大于也不影响结果
    if(b==0) return a;//注意，一定是 b 为零时返回 a。
    else return Euclid(b,a%b);//将 b 提前为第一个参数，之后是二者的余数
```

另外，计算最大公约数时需要注意两个特殊情况：

（1） gcd(a,b)=gcd(|a|,|b|);（即 a，b 为负数时计算其绝对值即可）；

（2） gcd(a,0)=|a|;

• 模运算的几个需要注意的点：以 a mod b=n 为例

（1）首先要知道模运算 b 要求是正整数，即 11 mod (-3)是错误的；

（2）负数模正数时，结果为 0 到 b 的一个正数，所以-11 mod 3 结果为 1，而不是-2（结果可能跟 Windows 上的计算器不同，但是书上是这么写的）

（3）同余的写法和读法，“整数 a 和 b 是模 n 同余的”，记作“a≡b （mod n）”，意思是（a mod n）=（b mod n）

（4）最后一点就是模运算本身的一些性质：

☺在书的第 76 页，证明题中经常会用到，一定要敏感（其实就是加减乘三种运算可以跟模运算交换计算顺序，但是要注意最后补的那一次模运算不要漏掉）。

以加法为例：

$$(a+b) \bmod n=((a \bmod n) +(b \bmod n)) \bmod n$$

☺在书的第 78 页，式 4.5：

如果(a*b)≡(a*c) (mod n),那么 b≡c (mod n)成立的条件是：a 与 n 互素

我这里换了一种跟书上不同的表述方法，目的就是想强调，在等式两边约去 a 时是有条件的，而这个条件是大家经常忽略的一点，如果不说明直接约去的话，可能阅卷时会扣一些分数，所以这个条件一定要记住。

好了，有了以上这些知识，应付简单的模运算计算题就绰绰有余了(不过也可能不会考这么简单的题目)

• 模运算的乘法逆元

之所以这一点没有放在上一点模运算中是因为，乘法逆元这一概念特别重要，许多计算题都需要求。书上给出了一种求乘法逆元的方法，即扩展欧几里得算法，但是由于算法比较复杂，所以并不要求掌握。考试中求乘法逆元主要用“猜”：

先来说说我们需要什么，对于 b mod n 这个式子来说，我们需要求出一个整数 a，使得 a*b mod n=1。

再来说说怎么猜，就是简单地去试，第一步先用 n+1 去除以 b，看能不能整除，若能则 a 为 n+1/b，这里比较绕，需要好好理解一下为什么。若不能，继续，n 先乘 2，然后加 1，再去除以 b，看是否整除，若能，则 a 为 2*n+1/b。抽象一下就是把 n 从 1 到 n-1 分别乘一下，再去除以 b，哪个能整除，a 就为 x*n+1/b。（描述比较抽象，跟着做一次就明白了）

另外还需要说明一点，就是并不是所有整数模一个数都有乘法逆元，即对于 a mod n，a 存在乘法逆元的条件是 a 与 n 互素。这个还可以引申到上面提到的约去时要求 a 与 n 互素，其实道理是一样的，可以看一下书 78 页的描述。

1. **利用费马定理计算 $3^{201} \bmod 11$。**

费马定理的具体内容在之后证明题的部分会涉及，这里也不说明了，直接上公式：

$$\mathbf{a^{p-1} \equiv 1 \pmod p}\text{,}\mathbf{其中要求\ p\ 为素数且\ a\ 与\ p\ 互素}$$

$$\mathbf{a^{p} \equiv a \pmod p}\text{,}\mathbf{只要求\ p\ 为素数即可}$$

解：因为 3 和 11 互素，由费马定理易知，$3^{10} \bmod 11=1$；

又根据模运算的性质可知，

$3^{200} \bmod 11=(3^{10})^{20} \bmod 11=(3^{10} \bmod 11)^{20} \bmod 11=1^{20} \bmod 11=1$

故 $3^{201} \bmod 11=[(3 \bmod 11)*(3^{20} \bmod 11)] \bmod 11=3*1=3$

有了费马定理，计算过程特别简单。要注意的就是找准指数和模数之间的关系，正确构造费马定理的形式。

2. 利用费马定理，找一个位于 0 和 28 之间的数 x，使得 x^{85} 模 29 与 6 同余。

这道题也不是很难，解法如下：

解：所求 x 满足方程 $x^{85}\equiv 6 \pmod{29}$

因为 29 是素数，且 x 在 0 到 28 之间，所以 x 一定与 29 互素

（这里注意如果 29 不是素数，那么这道题目除了用费马定理算出的结果外可能还有其他的结果也符合要求，不过考试应该不会涉及）

又 $x^{85} \bmod 29 =[(x^{28})^3 \pmod{29} *x \bmod 29] \pmod{29}$

$=(1*x) \pmod{29}=x=6$

故解得 x 值为 6。

3. RSA 相关计算

已知 p=17,q=11,e=7, M=88，求公钥 KU 和私钥 KR 分别为多少？加密计算后所得到的 C 为多少？并验证解密运算后，是否能恢复出明文 M。

说明：RSA 计算方法相对比较固定，所以只要自己亲自动手算几遍就能掌握了。

解：

n=p*q=187;

$\Phi(n)=(p-1)*(q-1)=160$;

$d=7^{-1} \bmod 160=23$;

所以，计算所得公钥 KU 为(7,187),计算所得私钥 KR 为(23,187)

$C=M^e \bmod 187=11$

$M=C^d \bmod 187=88$

看着计算过程，我们再具体说一下 RSA 的步骤。

首先，我们任意取两个素数 p 和 q，之后计算 n=p*q 和$\Phi(n)=(p-1)(q-1)$，（注意 n 应该>M）接着我们选择一个跟$\Phi(n)$互素且小于$\Phi(n)$数 e，作为公钥的第一个数，则公钥就能表示为(e,n)，之后我们计算 e mod n 的乘法逆元 d，计算方法见计算题，并将私钥表示为(d,n)。至此，我们已经计算出了加解密需要的东西，就可以进行加解密了。

加解密过程：$C=M^e \bmod n$ (用公钥加密)，$M=C^d \bmod n$（用私钥解密），要注意题中要求加密还是签名，签名就是私钥加密，公钥解密。

4. Diffie-Hellman 密钥交换

已知 q=353，α=3，A 和 B 分别选择密钥 X_A=97，X_B=233，求二者交换密钥的过程。

解：

A 和 B 首先计算各自的公钥，

$Y_A=\alpha^{X_A} \bmod q=40$；

$Y_B=\alpha^{X_B} \bmod q=248$；

则 A 产生的公钥可以表示为(q, α, Y_A)，B 则为(q, α, Y_B)

双方交换公钥后各自计算密钥 K，结果应该相同。

$K_A=(Y_A)^{X_B}=160$;

$K_B=(Y_B)^{XA}=160$;

说明：

q 是素数，α是其的一个本原根（定义在书 216 页上半部分），

要求 X_A、X_B 小于 q

5. ElGamal 密码体系

已知 q=19, α=10,XA=5,k=6,M=17，求两人交换信息的过程。

解：

首先接收者计算公钥，$Y_A=\alpha^{XA} \bmod q=3$；

接着发送者利用接受者公布的公钥对自己的信息进行加密。

发送者选择的随机整数 k=6，计算一次密钥 $K=(Y_A)^k \bmod q=7$;

则 $C_1=\alpha^k \bmod q=11$;$C_2=KM \bmod q=5$;

发送者发送的加密信息即可表示为(C_1,C_2)

对于接受者，首先恢复 K，$K=(C_1)^{XA} \bmod q=7$;

接着计算明文，$M=(C_2K^{-1}) \bmod q=(5*11) \bmod 19=17$;

说明：

虽然 ElGamal 和 DH 密钥交换都是基于离散对数的，但是 ElGamal 在计算过程上明显要比后者复杂，所以还需要重点掌握。

6. ElGamal 数字签名方案计算

已知 q=19，α=10，X_A=16，m=14，K=5，求签名和验证的过程。

解：

$Y_A=\alpha^{XA} \bmod q=4$

$S_1=\alpha^K \bmod q=3$

$K^{-1} \bmod (q-1)=11$

$S_2=K^{-1}(m-X_AS_1) \bmod (q-1)=4$

(

老师 PPT 上给出了另一种计算 S_2 的方法，即

$m=(X_A*S_1+K*S_2) \bmod q-1=14$

化简得：$(48+5S_2) \bmod 18=14$

即：$5S_2 \bmod 18=2$

$S_2=4$

总的来说，这种方法比较巧妙，但是计算过程中容易出错，所以建议对模运算性质掌握较好的同学使用

)

则生成的签名为(S1,S2)

验证时，$V_1=\alpha^m \bmod q=16$

$V_2=(Y_A)^{S1}(S_1)^{S2} \bmod q=16$

说明：

虽然这个算法也叫 ElGamal，但是可以看到它跟之前的算法有很多不同的地方，所以一定也要重点记忆。算法执行过程中有几点需要注意的地方，首先是 K 和 m 的选择，K 需要与 q-1 互素，m 需要在 0~q-1 之间，其次是计算 K^{-1} 和 S_2 时，模的都是 q-1，最后就是验证时，V1 和 V2 相等才算验证成功。

至此，所有教材中讲到的计算题都已经给出了计算步骤，接下来就是找一些

例题，加强自己的熟练度了。

证明题需要理解的东西比较多，需要的复习时间也比较长，因为理解起来也需要过程，而且如果想拿高分，不仅书上的要会，题目稍加变化也应该能做到正确分析和证明：

证明题部分也需要一些预备知识，即一些常用的证明思路，这样在证明一些没见过的命题时自己也有解决方法。

• 一个数的模运算表示：

在证明跟模运算有关的命题中，经常要将一个数表示为 k*b+r 的形式，即如果证明中用到了 a mod b，就可以把 a 表示为上述形式，之后再结合模运算的性质（计算题预备知识中有涉及）就可以了。

• 集合置换型的证明：

正确的名字不知道是什么，我们只需要知道要证什么和怎么证就行了。

要证的东西很简单，就是现在有两个集合，我们要证明这两个集合的所有元素都相同，只是其中元素的排列顺序不同。即{1,2,3}和{2,1,3}这种关系。

证明需要分三步，第一步是两个集合中元素的性质是相同的，即 B 中元素满足的性质同 A 中元素的相同；第二步要说明集合 B 中的元素互不相同（一般集合 A 是已知的，所以对于集合 A 一般不做分析，重点是讨论你构造出来的集合 B 是否满足性质）；最后一步就是，集合 B 中元素的个数与集合 A 相同。有了以上三点的约束，我们就能说明集合 B 是集合 A 的一个排列（跟之前的意思是一样的），之后的例题会具体说明怎么证明置换型。

1. **模运算性质证明：[(a mod n)*(b mod n)] (mod n)=(a*b) mod n**

证明：

设 $a=i*n+r_a$，$b=j*n+r_b$

则等式左边=(r_a*r_b) mod n

等式右边=$(i*j*n*n+i*n*r_a+j*n*r_b+r_a*r_b)$ mod n

=(r_a*r_b) mod n

=左边

说明：

将一个数表示成 i*n+r 的形式很重要。

2. **欧几里得算法（求最大公约数）证明：**

gcd(a,b)=gcd(b,a mod b)

算法的执行的详细过程在之前的计算题部分已经说过，这里就不再赘述了，直接来证明：

首先我们把证明转换一下，证明两个集合是相等的，即设集合 A 是 a 和 b 公因子的集合，集合 B 是 b 和（a mod b）公因子的集合，若集合 A 和集合 B 相等，则两集合中最大的数必然相等，即所证命题成立。

（因为 gcd（a,b）=gcd（b,a）,所以在证明时不妨设 a>=b，这样好理解一些，也可以不显式地说明）

（另外，这里跟书上的证明方法有所不同，我们有理由怀疑书上的证明是值得商榷的，它设的是 d=gcd(a,b)（即 d 是集合 A 中最大的元素）,然后证明 d 是 A 和 B 集合中的元素，显然这种方法只讨论了一个元素，并不能说明集合 A 和 B 是相等的）

设 d 为集合 A 中的一个元素，则 d|a 且 d|b。
而对于数 a，它可以表示为：a=kb+r,则 a mod b =r=a-kb。
因此，(a – kb)/d=a/d-kb/d，
因为 d|a,d|b，所以 d|(a-kb)，即 d|(a mod b)
故 d 也是集合 B 中的一个元素。

反之，如果 d 是集合 B 中的一个元素,则 d|(a mod b)
进而有,d|(kb+ a mod b)即 d|(kb+r)
所以 d|a
故 d 也是集合 A 中的一个元素。

(目前为止，我们已经证明了上面那个式子是成立的，但是如果要完整地证明欧几里得算法，我们还需要证明通过多次迭代之后求出的结果就是最大的公因子)

注意到欧几里得算法迭代停止的条件为 b=0，由定理可知，
gcd(a,0)=|a|,即最后所得的数就是数 a 和 0 的最大公约数，也就是最初要求的最大公约数。
命题得证。

3. 使用欧几里得算法来证明：

简单地说，之前的欧几里得算法只是一个工具，我们还可以利用这个工具来证明其他的延伸命题:

对两个连续整数 n 和 n+1，有 gcd(n,n+1)=1。

证明： 由欧几里得算法可知，
gcd(n,n+1)=gcd(n+1,n)=gcd(n,n+1 mod n)=gcd(n,1);
而任何数与 1 的最大公约数只能是 1，故命题得证。

4. 费马定理（即费马小定理）

命题:若 p 是素数，a 是正整数且不能被 p 整除，则

$$a^{p-1} \equiv 1 \pmod{p}$$

（或表述为：若 p 是素数，a 是正整数，则 $a^p \equiv a \pmod{p}$）

说明：其实 p 是素数和 a 不能被 p 整除这两点还能说明，a 和 p 是互素的（有的同学可能只注意到了第二个条件，就会误以为 a=9 和 p=6 也满足上面的等式，但是 6 并不是素数，而且 6 和 9 也不互素，所以这种情况不能用费马定理）

思路：首先我们需要构造一个集合 X，然后证明置换型，最后将其中的元素相乘就能证明这个定理啦。具体过程如下。

证明： 定义一个集合 X 为{1*a mod p,2*a mod p,…,p-1*a mod p}，
考虑另一个集合 A{1,2,3,…,p-1}，我们说 X 是 A 的一个置换型。
置换型证明：
首先因为 p 是素数，且 a 与 p 互素，集合 X 中没有元素的值为零
两个集合元素范围均在 1~p-1 之间（A 和 X 元素性质相同）
考察 X 中的任意两个元素 i*a mod p 和 j*a mod p(1<=i<j<=n-1,即 i!=j)
若两个元素的值相等，则 i*a mod p=j*a mod p
因为 a 和 p 互素，则可以约去 a，即 i mod p=j mod p

有 i 和 j 都是小于 p 的，所以 i=j，这与我们的假设矛盾
所以集合 X 中所有元素均是不同的（X 中元素各不相同）
显然，集合 X 中元素如果各不相同，
则集合 X 中元素的个数也为 n-1 个（A 和 X 元素个数相同）
所以，X 是 A 的一个置换型。
故将 X 中的元素相乘的结果应该与 A 中元素相乘的结果相同，即
(a*2a*3a*…*(n-1)a) mod p=(1*2*3*…*p-1) mod p=((p-1)!) mod p
等式左边可化为:[$a^{(n-1)}$ mod p *((p-1)!) (mod p)] mod p
因为 p 是素数，所以(p-1)!与 p 互素，所以可以约去(p-1)!，
所得结果即为所证。

（这里我说得比较详细，其实如果大家能把书上证明看懂，我这个自然也就懂了）
最后说明一点，就是费马定理的另一种形式，$a^p \equiv a \pmod p$，它不要求 a 与 p 互素，是因为如果 p 是素数，且可以整除 a，则 a 一定是 p 的倍数，故 a mod p=0，以上定理是在这种特殊情况下也成立的。

5. 欧拉函数的证明：

首先我们需要知道欧拉函数$\Phi(n)$指的是小于 n 且与 n 互素的正整数的个数。
根据定义，显然若 p 是素数，则$\Phi(p)=p-1$。
有了上面亮点，我们来看需要证明的东西
命题：设 p 和 q 是两个素数，且 p!=q，则对于 n=p*q 来说，

$$\Phi(n)=\Phi(p)*\Phi(q)=(p-1)*(q-1)$$

说明：首先说明的是小于 n 且与 n 互素的正整数中是包括 1 的，
即$\Phi(1)=\Phi(2)=1$，其中$\Phi(1)$就是特殊情况了。
其次是一定不要忽略 p!=q 这个条件，书的 178 页有计算好的欧拉函数值表，可以参照表格去验证一下（$\Phi(9)=6!=\Phi(3)*\Phi(3)=2*2=4$）
证明：
考察{1,2,3,…,p*q-1}这个集合，集合中与 n 互素元素的个数即为$\Phi(n)$。
显然，这个集合中与 n 不互素的元素的集合为{p,2*p,3*p,…,(q-1)*p}和{q,2*q,3*q,…,(p-1)*q}。
设集合 A 为{p,2*p,3*p,…,(q-1)*p},集合 B 为{q,2*q,3*q,…,(p-1)*q}
显然，集合 A 中元素个数为 q-1 个，集合 B 中元素个数为 p-1 个，且集合 A、B 中所有元素均不相等。
因为若 i*p=j*q,则 p/q=j/i，因为 p、q 互素，则 p/q 不能约分，所以 j 和 i 的值必须是 p、q 值相同的倍数，这与假设矛盾，所以集合 A、B 中所有元素均不相等。
故$\Phi(n)=(p*q-1)-(p-1)-(q-1)=p*q-p-q+1=(p-1)*(q-1)=\Phi(p)*\Phi(q)$,得证。

6. 欧拉定理的证明：

有了欧拉函数，我们就能继续讨论欧拉定理了，它的证明思路类似于之前费马定理的证明，所以有了前面的基础不难证明这个定理。
命题：对于任意互素的 a 和 n，有

$$a^{\Phi(n)} \equiv 1 \pmod n$$

(第二种形式，对于任意数 a 和 n，$a^{\Phi(n)+1} \equiv a \pmod n$)

说明：证明前需要明白欧拉定理与费马定理的区别，那就是指数这里是$\Phi(n)$，而不是 n-1，说实话这个定理在实际计算中似乎没有什么用处，我们只是用它来理解之后公钥算法的原理。

证明：

设集合 R 为$\{x_1,x_2,\ldots,x_{\Phi(n)}\}$,其中 x_i 表示第 i 个小于 n 且与 n 互素的正整数。

设集合 S 为$\{ax_1 \pmod n, ax_2 \pmod n, \ldots, ax_{\Phi(n)} \pmod n\}$，则 S 是 R 的一个置换型。

首先，a 与 n 互素，x_i 与 n 互素，则 ax_i 也与 n 互素，故 S 和 R 中元素的性质相同。

其次，若 $ax_i \pmod n = ax_j \pmod n$,则因为 a 与 n 互素，可约去，

的 $x_i=x_j$，这与假设矛盾，所以集合 S 中元素各不相同。

显然，若元素不同，则 S 中元素个数与 R 相同，故 S 是 R 的一个置换型。

因此，将 S 和 R 中的元素分别累乘结果相同，得：

$(ax_1 \bmod n)*(ax_2 \bmod n)*\ldots*(ax_{\Phi(n)} \bmod n)=(x_1*x_2*\ldots x_{\Phi(n)})$

等式左侧可化为$[(a_{\Phi(n)} \bmod n)* (x_1*x_2*\ldots x_{\Phi(n)}) \bmod n](\bmod n)$

因为 x_i 是与 n 互素的，所以其乘积也与 n 互素，故可约去，得

$a_{\Phi(n)} \equiv 1 \pmod n$，得证。

(至于第二种形式，证明目测比较复杂，因为定理没有要求 n 是素数，所以略去)

7. 中国剩余定理(CRT)的证明

中国剩余定理是考试中比较常考的一个点，所以掌握其内容和证明是基本要求。

命题：先形式地陈述一下，CRT 说明某一范围内的整数可通过它的一组剩余类数来重构，这组剩余类数是对该整数用一组两两互素的整数取模得到的。

说明：看了上面的定义，有好多术语不是很熟悉，所以有必要再通俗地说明一遍，CRT 到底能干什么。

先看之前的那道计算题

今有物不知其数, 三三数之剩二, 五五数之剩三, 七七数之剩二, 问物几何。

列出方程即：

$x \bmod 3=2$	$n=3*5*7=105$
$x \bmod 5=3$	$d_1=3, d_2=5, d_3=7$
$x \bmod 7=2$	$x_1=2, x_2=3, x_3=2$

CRT 说明，若 d1，d2，d3 这三个数两两互素，则在 n 这个范围内，能唯一求出 x，同时满足上面的三个方程。

看了描述，我们知道 CRT 其实是用来解方程组的，方程组长这样：

$$(S):\begin{cases} x \equiv a_1 \pmod{m_1} \\ x \equiv a_2 \pmod{m_2} \\ \vdots \\ x \equiv a_n \pmod{m_n} \end{cases}$$

要求 m_1 到 m_n 是两两互素的，而我们能算的就是 x，证明的话就是要说明我们在 $0\sim m_1*m_2*\ldots*m_n$ 这一范围内能够求出唯一的解 x，并且这个解是正确的（但是唯一性不要求证明）

完整的证明分为两步，第一步没有什么技巧，需要牢记定理构造解的方法（其实这也属于定理的一部分），第二步是证明第一步中得到的答案是正确的。

证明：

令 $M=m_1*m_2*m_3*\ldots*m_n$,其中 m_i 两两互素

$M_i=M/m_i$（这样 M_i 就与 m_i 互素了）

再另 $c_i=M_i*(M_i^{-1} \bmod (m_i))$（即 c_i 是 M_i 和其模 m_i 乘法逆元的乘积，注意此时还未求模，如果求模 m_i 后结果才是 1）

则最后的 x 可以表示为 $x=(a_1c_1+a_2c_2+\ldots+a_nc_n)(\bmod M)$（注意 a_i 是原式中的余数，第一步证毕）

接下来验证我们构造的解是否符合条件：

要证明上述命题，我们要求 x 满足对于所有 i(1<=i<=n)，

有 $a_i=x \bmod m_i$

因为当 j!=i 时，$M_j \bmod m_i=0$，即 $c_j \bmod m_i=0$，

所以当 $x \bmod m_i$ 时，只需考察 a_ic_i 项

而 $c_i \bmod m_i=1$，所以 $x \bmod m_i$ 结果为 a_i，得证。

（说明一点，这里比较难理解的就是为什么 $c_i \bmod m_i$ 结果为 1，我们举个例子来看一下：若 M_i 为 6，m_i 为 7，则 c_i 为 6*6=36，这时还未模 m_i，所以结果不是 1，等最后计算时模 m_i 之后，结果就是 1 了。形式化描述一遍就是：$(a^{-1}*a) \pmod n=1$，而 $a^{-1}*a$ 不是 1）

8. 证明：n>2 时φ(n)是偶数。这一点对所有 n>2 都成立。

说明：题目出自老师课堂上的一次练习。老师说他上课几年中，还没有学生做出来过，而且考试也不考这么难的，所以就直接贴答案，然后简单说下思路了。

思路：整体思路就是证明 n>2 时，计算在ϕ(n)中的数总是成对出现的。为了证明这一点，有引出两个分论点，第一个是 gcd(a, n) = gcd(n – a, n)，第二个是证明 a 和 n-a 是不同的两个数。

证明：

假设 a 是被计算在ϕ(n)中的一个整数，则 n-a 为另外一个被计算在φ(n)中的整数，因为 gcd(a, n) = gcd(n – a, n). 假设 n-a 和 n 有最大公因子 b，且 b≠1，则有 xb=n-a, yb=na=n-xb, n=yb—>a=yb-xb=(y-x)b，可以推导出 gcd(a,n)=b。同时，这两个整数 a 和 n-a 是不同的，因为如果相同，则有 a=n-a，即 n=2a，那么 gcd(a,n)=a。因此，当 n>2 时，φ(n)中数是成对出现的，所以必是偶数。

9. RSA 密码体制原理的证明

先给出一种错误的证明

错误定理：

给定两个素数 p、q，令 n=p*q，$ed \bmod \varphi(n) =1$, $m\in[0, n-1]$,

gcd(m, n)=1,

则：$(m^e \bmod n)^d \bmod n = m^{ed} \bmod n = m$

证明：

因为 $ed \bmod \varphi(n)=1$,所以 ed 可表示为 $k\varphi(n)+1$

故 $m^{ed} \bmod n = m^{k\varphi(n)+1} \bmod n = m*(m^{k\varphi(n)}) \bmod n = m*(m^{\varphi(n)})^k \bmod n$

由欧拉定理 $a^{\varphi(n)} \equiv 1 \pmod n$（a 与 n 互素）易知

原式= $m \bmod n = m$。

错误原因：

RSA 在加密过程中不能对密文有所要求，即错误定理中的 gcd(m,n)=1 这个条件不是恒成立的，所以虽然上述证明过程是正确的，但由于命题本身加强了条件，所以证明整体上有所欠缺。

正确命题：

给定两个素数 p、q，令 n=p*q，ed mod φ(n) =1, m∈[0, n-1]

则：$(m^e \bmod n)^d \bmod n = m^{ed} \bmod n = m$

证明：

(1) 若 m 与 n 互素，则证明过程同之前的错误证明；

(2) 若 m 与 n 不互素，则证明前还需要知道一个由 CRT 推导出来的定理，即

若 a≡b mod pq，当且仅当 a≡b mod p 且 a≡b mod q

那么我们要证 $m^{ed} \bmod n = m$，即可分别证 $m^{ed} \bmod p = m \bmod p$、$m^{ed} \bmod q = m \bmod q$

以证 $m^{ed} \bmod p = m \bmod p$ 为例

因为 m 与 n 不互素，所以 $m=x_1*p$ 或 $m=x_2*q$(即 m 与 p 不互素或互素)

(a) 若 m 和 p 不互素，因为 p 是素数，则 p 可整除 m，故 m mod p=0；所以 $m^{ed} \bmod p=0=m \bmod p$;

(b) 若 m 和 p 互素，则 gcd(m,p)=1

$m^{ed} \bmod p=m^{k*(p-1)(q-1)+1} \bmod p=(m*(m^{(p-1)} \bmod p)^{k(q-1)}=m \bmod p$

同理可证：$m^{ed} \bmod q = m \bmod q$

命题得证。

说明：由 CRT 推导出来的那个公式是网络安全专业要求的公式，并不是软件工程专业需要掌握的，但是即使不知道那个公式，我们也能得到正确的结论。证明时还是先证 $m^{ed} \bmod p = m \bmod p$，同理得 $m^{ed} \bmod q = m \bmod q$，之后我们证明$(m^{ed}-m) \bmod p=0$（利用模运算性质），同理得$(m^{ed}-m) \bmod q=0$，即数 $m^{ed}-m$ 可同时被 p 和 q 整除。所以 $m^{ed}-m=r*p*q=r*n$，即 $m^{ed}-m$ 也可被 n 整除，故$(m^{ed}-m) \bmod n=0$，也就能推出 $m^{ed} \bmod n=m \bmod n$，命题得证。

10. 离散对数相关证明

关于离散对数的证明其实不难，重点记忆的还是它的加解密过程，知道过程后自然能够推导和证明出解密的正确性。

这里给出一个最基础的证明，就是 D-Hellman 密钥交换的证明。

命题：

根据 D-Hellman 密钥交换的过程证明通信双方计算出的 K 值是相同的。

证明：

通信方 A 计算 $K=(Y_B)^{XA} \bmod q=(\alpha^{XB} \bmod q)^{XA} \bmod q= (\alpha^{XB})^{XA} \bmod q=\alpha^{XAXB} \bmod q =(\alpha^{XA})^{XB} \bmod q=(\alpha^{XA} \bmod q)^{XB} \bmod q =(Y_A)^{XB} \bmod q$=通信方 B 计算。

说明：

完整地写出证明过程还是比较不容易地，所以一定要自己再熟悉一遍证

明过程，每一步的原因都需要知道。

分析题(两道)

分析题出题范围也比较广，主要考察学生对于某个具体过程的理解，而且分析题也会考一些之前书本上没有的知识点，要求学生学会灵活变通，但虽然考点发散，但是总的思路和方法是不变的，所以分析题也有规则可循。还是先来说说预备知识：

• 公钥密码学中消息的保密和认证：

非对称密码可以用来保密、认证或者两者兼而有之。保密是用对方的公钥加密，而认证是用自己的私钥加密。保密和认证兼具时要求先用对方公钥再用自己私钥加密。

• 一些记号必须知道，比如如果字符上面有^记号，表示的是攻击者试图恢复的内容，还有一些记号见下表

……

1. DES 加密过程分析

对 DES 加密过程的分析主要是根据图来分析的，首先要知道 DES 的分组及密钥的位数、DES 迭代的轮数和经典的 Feistel 结构，然后才能进一步理解 DES 加密的过程(这些在选择和简答里都有涉及)。先来看两个图：

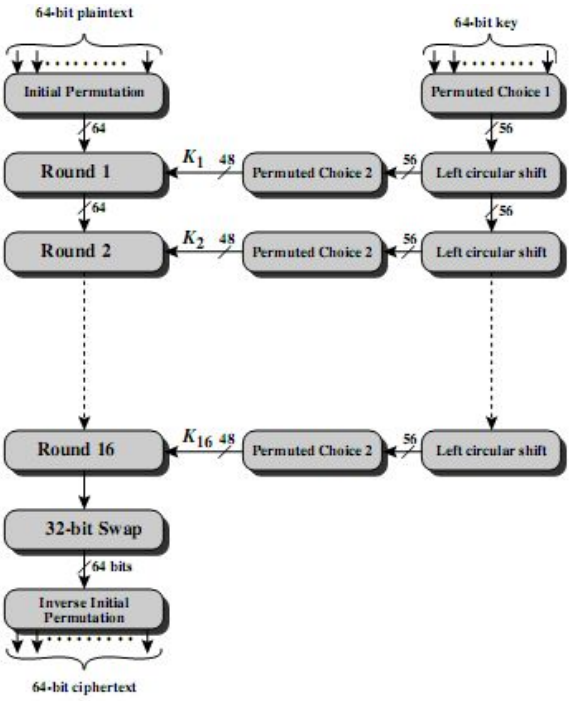


Figure 3.4 General Depiction of DES Encryption Algorithm

图 1 DES 加密算法整体描述

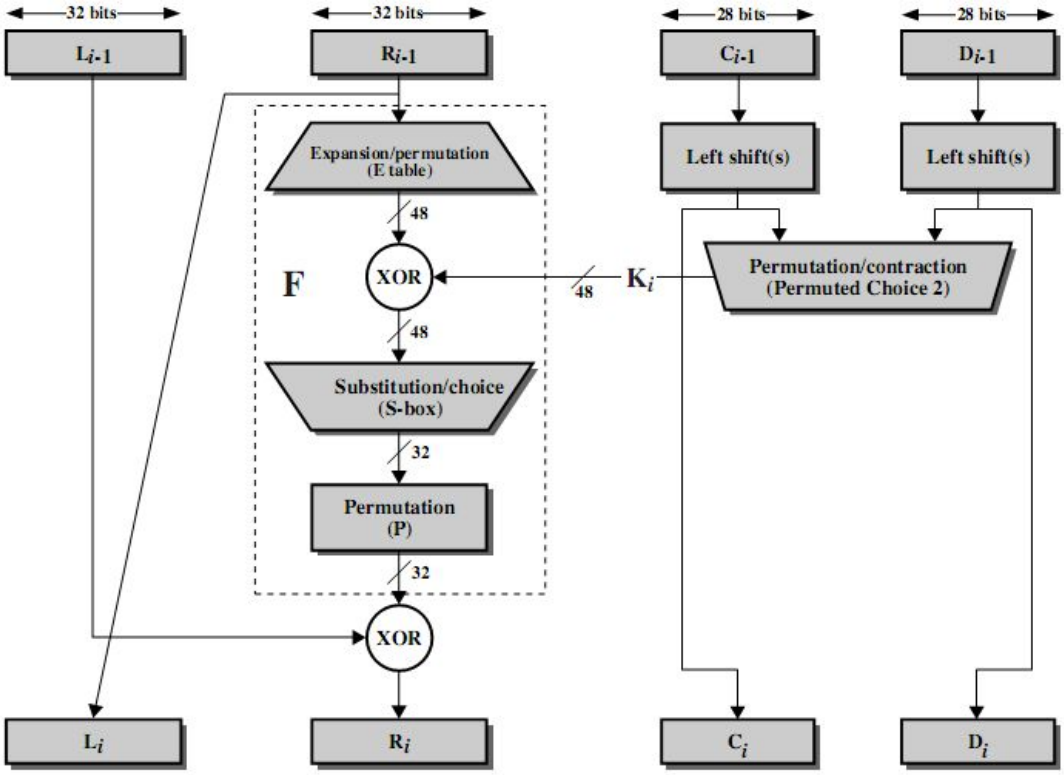


Figure 3.5 Single Round of DES Algorithm

图 2 DES 算法一轮迭代的过程

图 1 描绘了 DES 算法的整体过程，首先它对 64 位的明文做了一个初始置换(用的是矩阵)，然后迭代 16 轮，最后在进行一次初始置换的逆置换产生 64 位的密文。

这里重点介绍一下图 2 中描绘的每一轮迭代的具体过程。首先 64 位的明文输入分为左右两部分 L 和 R，各 32 位。然后将 R 通过一个 **E 表(E 置换)**，扩展成 48 位，注意这里是第一个重点，它的作用是**扩散**。这时，这一轮的子密钥同样也会生成，其长度也为 48 位，二者进行异或之后送入 S 盒。这里是第二个重点，**S 盒(S 置换)**实现了**混淆**的目的，其变化是 DES 中唯一的**非线性变换**(即对 DES 的安全性完全基于 S 盒，攻击的重点也在这里)。S 盒的输出又使数据长度变回了 32 位，最后再进行一次 **P 置换**(又称 P 盒，注意这里网上有文章说 P 盒才是 DES 实现扩散的方法，到底哪个正确按授课老师所说为主，但是 DES 的先扩展再压缩一定是为了实现扩散这一目的)，与开始的 L 再进行一次异或，就完成了这一轮的迭代（下一轮的 L 是这一轮的 R，下一轮的 R 是这一轮最后的结果，看图理解就行）

至于子密钥的生成，这里就不再赘述了，主要记住其生成方法是置换和压缩，然后将两个 28 位的 L 和 R 变成 **48 位的子密钥**即可。

2. 公钥授权分配方案的分析

这里对应的知识点为公钥密码体制三大应用中对密钥分配问题的应用，这里分配的密钥即公钥密码体制中的公钥。先上图：

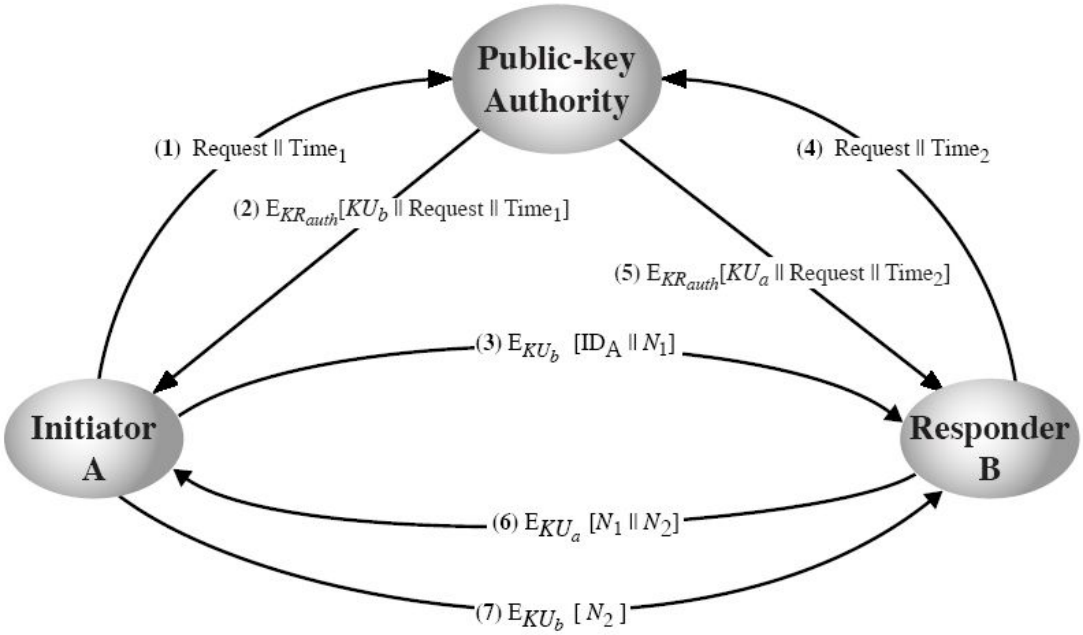


Figure 10.3 Public-Key Distribution Scenario

考试不会要求你默写出整个图来，只会给出图，然后问你每一步具体的操作，然后说明一下这步操作的意义(即满足了什么条件，能防止什么攻击之类的)。下面一步步来介绍：

(1) 密钥申请方 A 向中心发出一条带有时间戳的申请，请求 B 的公钥；

(2) 管理员给 A 一条用自己私钥加密的信息，里面包含 B 的公钥，以及 A 刚才发出的申请。**注意，这里出现了考点，第一点**就是管理员用私钥加密的作用是，能够让 A 确信该消息确实来自管理员；**第二点**是管理员在信息中包含了 A 的原始请求，可以让 A 确信其请求未被篡改；**第三点**是管理员在信息中包含了 A 的时间戳，可以让 A 确信自己收到的消息不是之前的旧消息，防止重放攻击；

(3) A 用 B 的公钥加密自己的身份标识 ID_A 和临时交互号 N_1，发送给 B，其中 N_1 唯一标识该次交互；

(4) (5)B 同(1)、(2)一样，得到 A 的公钥；

至此，通信双方已经都拿到了对方的公钥，可以通讯了，但是最好执行下面两步：

(6)B 用 A 的公钥加密 A 发来的临时交互号 N_1 和自己产生的 N_2，这样 A 解密后看到 N_1 即可确信消息来自 B，完成认证的过程；

(7)同样，A 将 N_2 加密后传至 B，B 解密后即可确信消息来自 A。

第二个考点，N_1、N_2 的作用，实现通信双方对身份的相互认证，保障了消息来源的真实性。

3. 公约证书分配方案的分析

这个方案跟上面的方案解决的是同一问题，但是由于证书方案比较复杂，所以下来说明一下证书方案的具体内容：

公钥证书将一个通信方的**身份**与他的**公开密钥**绑定在一起，通常还包括有效期和使用方法等；
证书的所有内容必须经由可信公钥授权方或者证书授权方**签名后**方可生效；
知道公钥授权当局公开密钥的任何人**都可以验证**一个用户的公开密钥证书的有效性；
对于申请者 A，管理员提供的证书为：$C_A = E_{KRauth}$ [T, ID_A, KUa]
(即用授权方私钥加密的时间戳||A 的身份||A 的公钥，所有拥有授权方公钥的通信方都可以进行解密得到 A 的公钥，**这里的考点主要在时间戳 T 上**，它的作用是验证证书的时效性，假设 A 的私钥泄露，并及时申请了新的证书，则旧证书的 T 失效，因此攻击者即使**重放** A 的旧证书，企图让 B 用 A 的旧公钥加密信息，那么攻击者也无法得逞。即时间戳 T 可以控制证书是否有效，如果私钥泄露，T 可以实现”挂失”功能，书上 307 页的例子很好)
接下来上图分析：

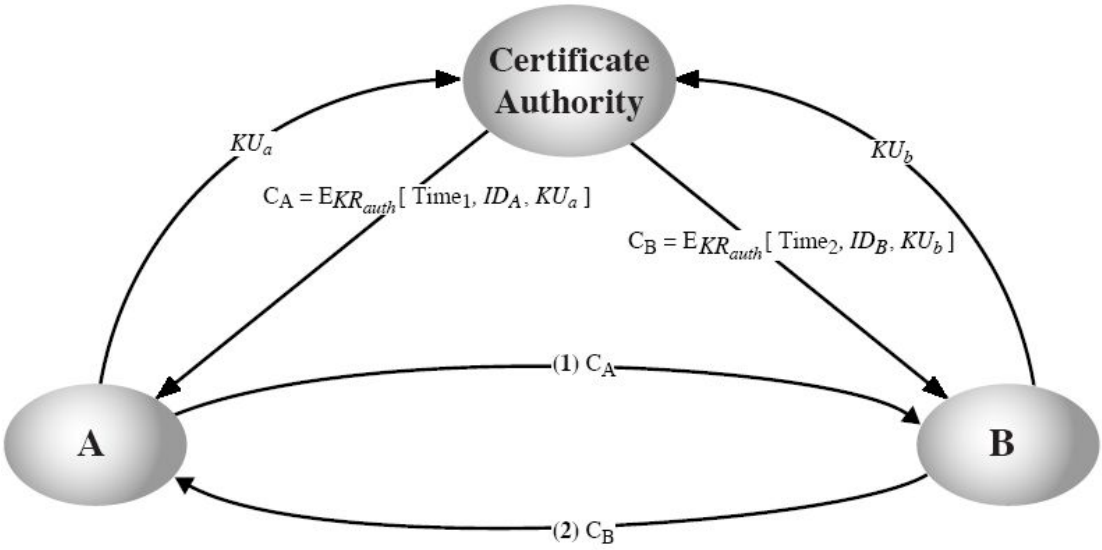


Figure 10.4 Exchange of Public-Key Certificates

首先，没有序号的四条线分别表示 A 和 B 单独向中心申请证书的过程，只分析 A：
A 向中心提交自己的公钥，请求得到自己的证书；
中心返回 A 的证书，这个证书跟之前的结构完全相同，不再赘述；
B 类似。双方拿到自己证书后互换，即可实现公钥的共享。
4. D-Hellman 密钥交换中的中间人攻击
之前我们讨论的问题是交换通信双方的公钥，现在我们来讨论如何在通信过程中实现共享一个相同的密钥K,这个密钥可以作为对称密码系统中的公钥。一种简单的思想就是 D-Hellman 的密钥交换协议，它的步骤在计算题中有涉及，现在重点讨论一下针对这个协议的一种特有的攻击手段，中间人攻击。
先上图：

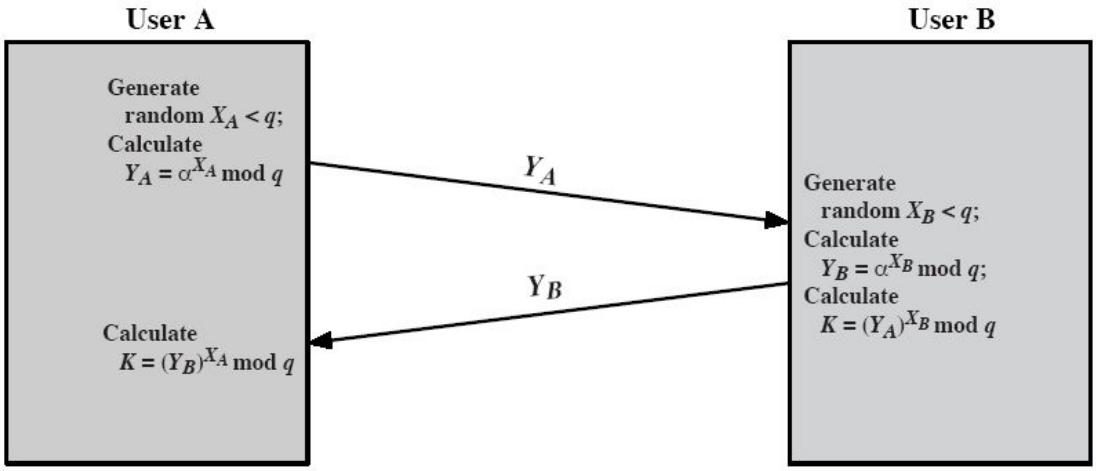


Figure 10.8 Diffie-Hellman Key Exchange

协议的正确性已经讨论过了，我们现在假设在 A、B 之间有一个 C。
首先，C 截获了 A 传来的 Y_A，模仿 A，C 也可以生成自己的 Y_A'，并传给 B；
同时 B 得到 Y_A'后，发送给 C 自己的 YB'，并且计算出 K'；
C 截获 YB'后即可算出与 B 共享的密钥 K'，实现与 B 的通信；
同时 C 也可以模仿 B，算出 Y_B，返回给 A，这样 A 和 C 又共享密钥 K；
至此，C 成为能够同时和 A 和 B 进行通信的人(通过两个不同的密钥 K 和 K')，而且 A 和 B 都不能察觉，即 C 完成了对协议的攻击。
另外，老师 PPT 上还给出了另外两种密钥交换的方法，这里也简单讨论一下：
第一种：**简单的秘密密钥分配**

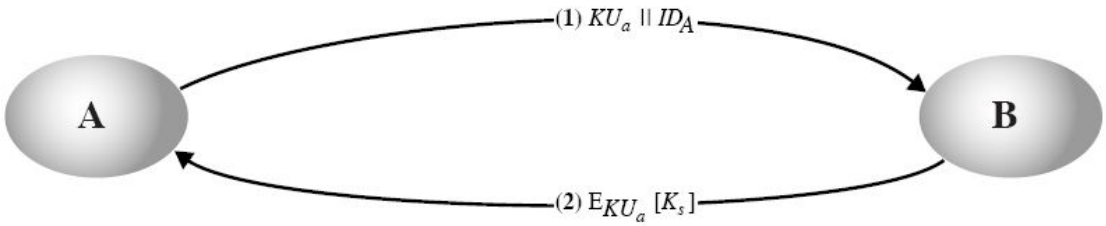


Figure 10.5 Simple Use of Public-Key Encryption to Establish a Session Key

本协议不能抵抗中间人攻击，很容易想象，如果 A 和 B 之间存在 C，那么他可以先拿到 A 的公钥，然后发给 A 一个 K，之后再将自己的公钥发给 B(IDA 不作替换)，得到 B 发来的密钥 K'，即实现了中间人攻击。
第二种：**具有保密性和真实性的密钥分配**

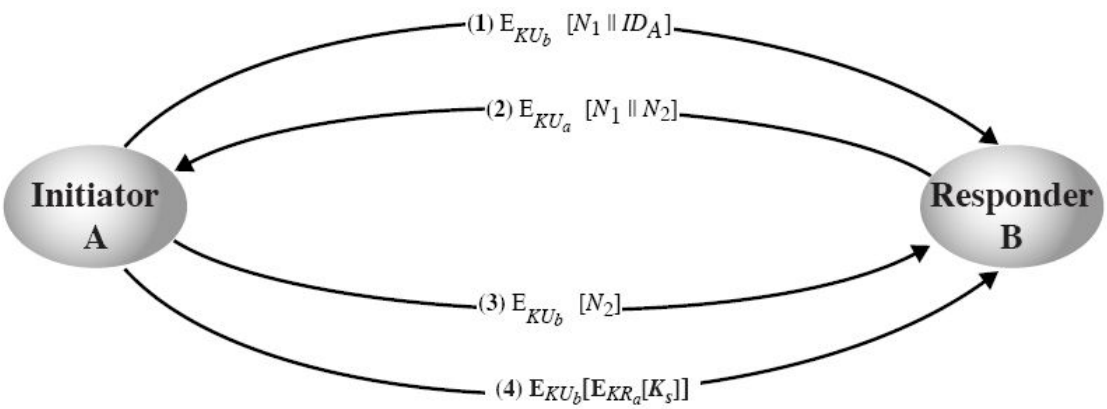
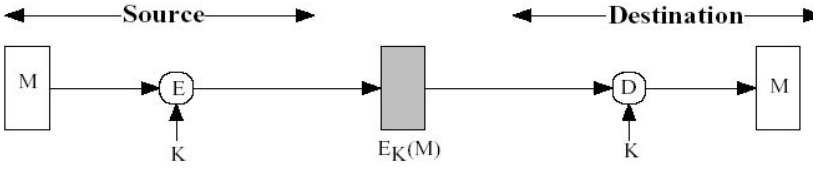


Figure 10.6 Public-Key Distribution of Secret Keys

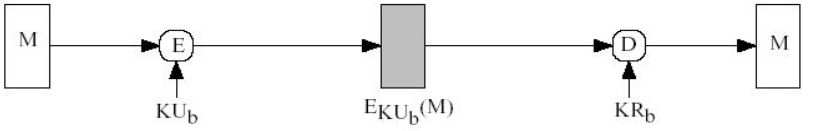
这种方案明显要比上一种方案要安全许多。首先，发送方 A 用 B 的公钥加密自己的请求发送给 B，然后 B 收到请求后进行回复，确认自己和对方的身份，接着 A 先确认收到，之后用自己私钥和对方公钥加密密钥 Ks，实现保密和认证目的。

5. 与 MAC 和 Hash 相关的问题分析

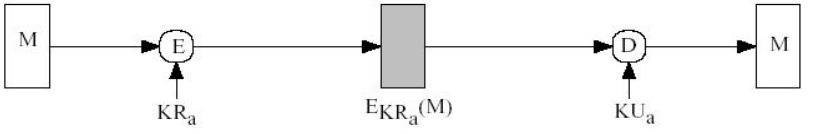
(0)在简答题部分，我们提到了可以用**对称密码或私钥**来实现认证，这里直接上四个图，来简单分析一下：



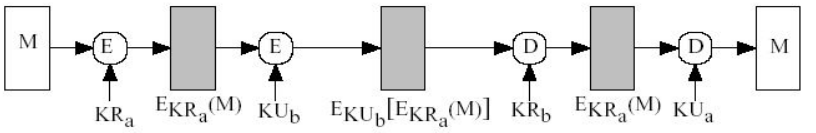
(a) Conventional encryption: confidentiality and authentication



(b) Public-key encryption: confidentiality



(c) Public-key encryption: authentication and signature



(d) Public-key encryption: confidentiality, authentication, and signature

第一种：用对称密码实现认证；
第二种：对方公钥加密，实现保密性；
第三种：自己私钥加密，实现认证性；
第四种：先用自己私钥再用对方公钥加密，保密性和认证性同时实现(之前也有涉及)。
这四种思想是比较基础的，不属于重点，我们这里重点讨论一下 MAC 和 Hash 在消息认证中应用及其相关的图都罗列一下，方便大家进行对比和总结：
(1)首先是 FCS(Frame Check Sequence)，FCS 类似于我们之前学过的奇偶校验码，可以在一定程度上对消息的正确性进行验证，根据填加 FCS 和执行加密函数的先后顺序可以分为两种：

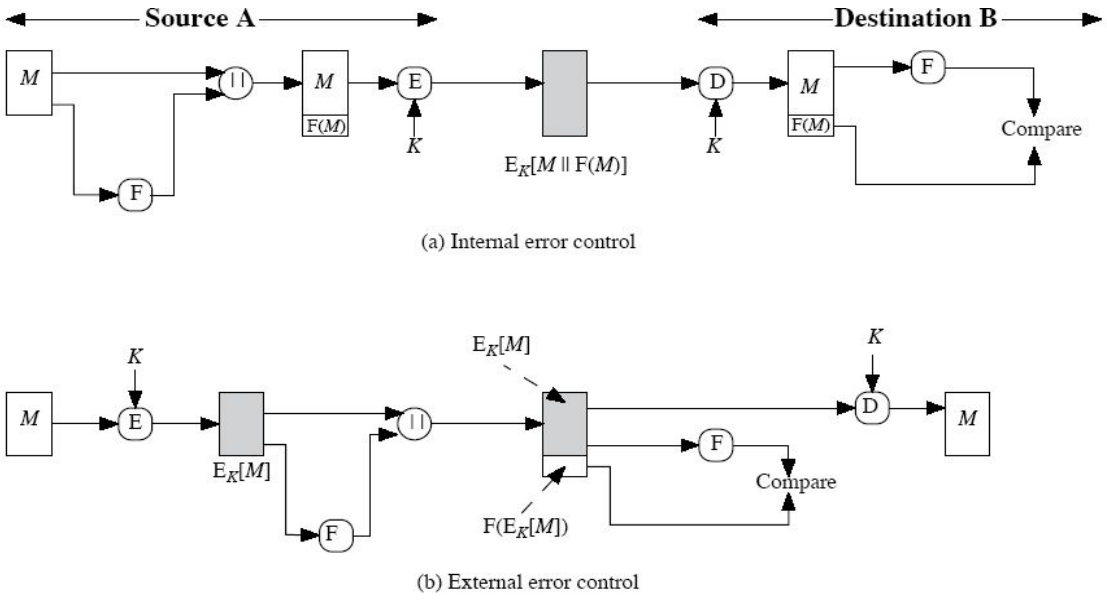


Figure 11.2 Internal and External Error Control

第一种：内部差错控制，先加 FCS 再加密，内部的意思就是对明文控制；
第二种：外部差错控制，先加密在对密文填加 FCS，外部指对整个密文进行控制。
(2)MAC(消息认证码)：通信双方 A 和 B 共享密钥 K，报文从 A 发往 B，A 计算 MAC=CK(M), 附在报文后发给 B。B 对接收到的报文重新计算 MAC，并与接收到的 MAC 比较。

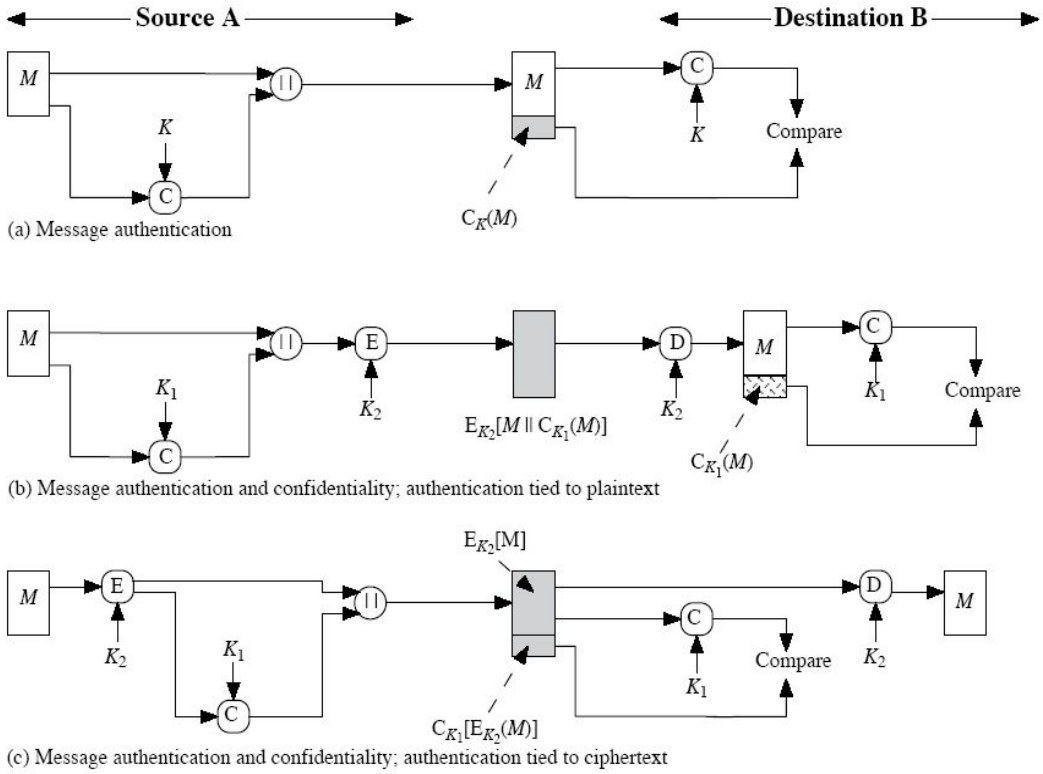


Figure 11.4 Basic Uses of Message Authentication Code (MAC)

第一种：用 MAC 实现消息认证，但消息是明文传送的；
第二种：用 MAC 实现消息加密和认证，即将第一种的消息再进行一次加密，这样传送的消息就是密文了，另外注意这里 MAC 是对明文进行的；
第三种：消息先进行加密后再使用 MAC 来认证，同样可以实现消息的保密和认证，但是这种方法是对密文进行 MAC，要注意和第二种的区别。
(3)Hash 散列函数：以变长的报文 M 作为输入，产生定长的散列码 H(M)，作为输出，但是加密时不需要密钥，因此使用更加灵活，这是它跟 MAC 重要的区别。

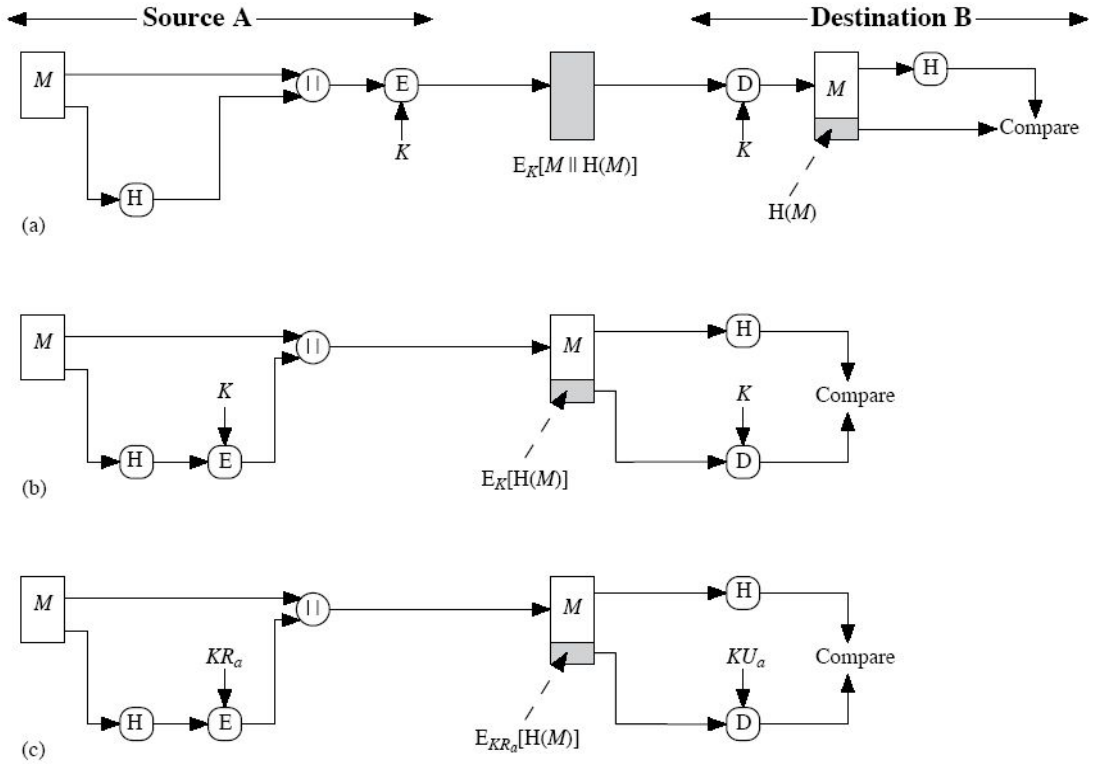


Figure 11.5 Basic Uses of Hash Function (page 1 of 2)

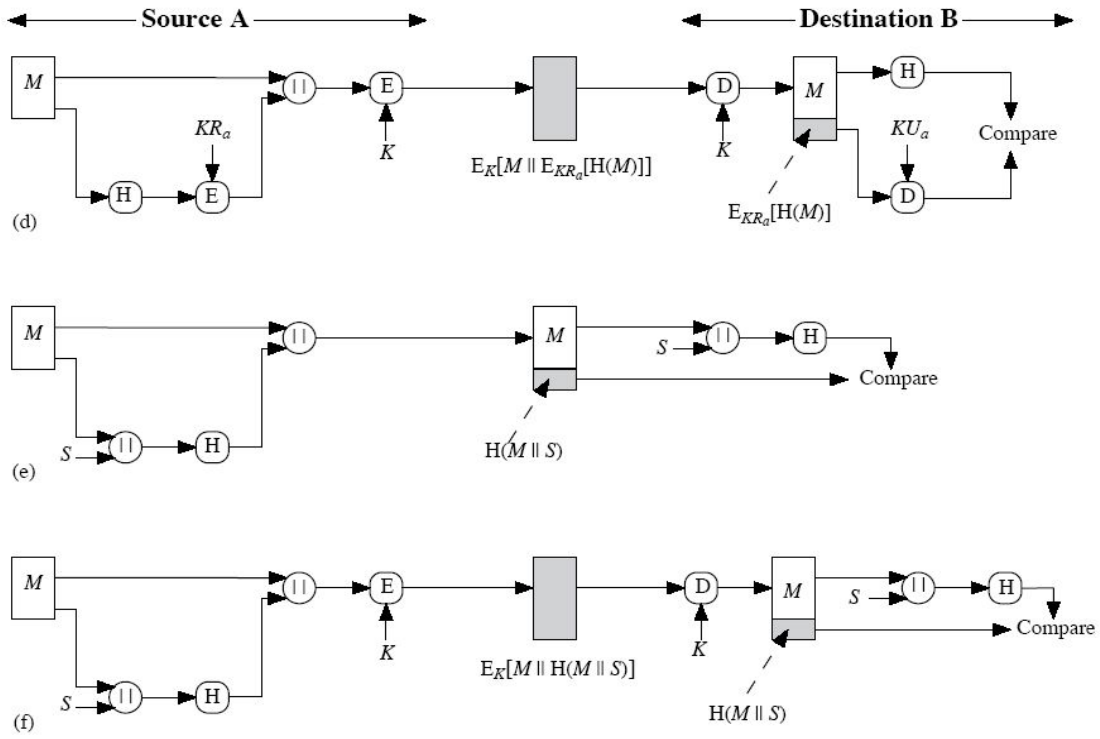


Figure 11.5 Basic Uses of Hash Function (page 2 of 2)

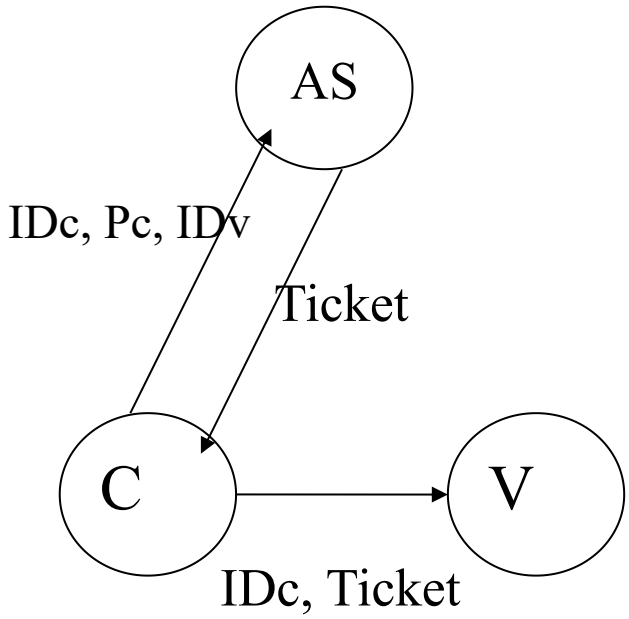
第一种：对明文哈希后整体加密；
第二种：对明文哈希后只对哈希函数的输出加密，消息本身没有加密，明文传送；
第三种：同上一种，只是在加密时用公钥密码体制，即对方的公钥加密；
第四种：加密两次，先是用对方公钥加密哈希值，然后在用对称密码加密整个消

息；
第五种：二者事先共享了一个秘密值 s，然后明文串上 s 之后进行哈希，将结果再与明文串，攻击者同样无法篡改信息；
第六种：上一种的基础上，对传送值进行一次加密。
上述六种方法**均能实现消息认证**，因为有对称和非对称密码体制的支撑，但是对于保密性，如果传送时对报文没有加密，则不能实现保密性。即使这样，攻击者最多只能阅读明文的信息，而不能私自篡改和伪造信息，从而实现了消息的认证。

6. Kerberos 认证服务分析

Kerberos 要求**客户向服务器提供身份认证，服务器向客户提供身份认证**，来保护用户信息和服务器资源,。它利用一个**可信的第三方**认证服务来完成客户端和服务器端的认证，而且**仅依赖于对称密码体制**(这里可与将要提到的 X.509 区别)，目前主要使用版本 4 和 5，虽然 5 比 4 有许多改进之处，但是我们还是重点介绍针对版本 4 的分析：

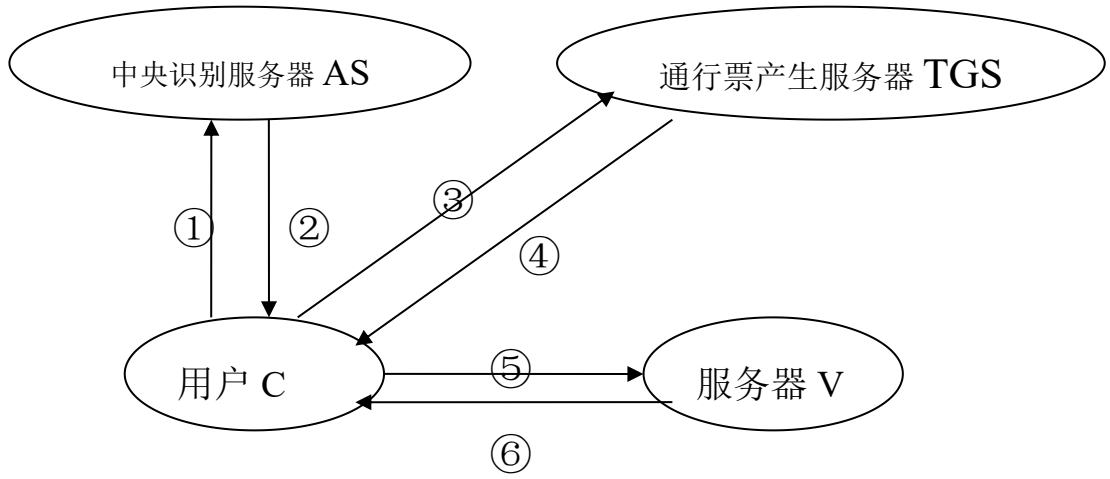
(1)**一个简单的认证会话:**



Ticket: $E_{KV}[ID_C, AD_C, ID_V]$
ID_C: 用户 C 名
ID_V: 服务器 V 名
AD_C: 用户 C 的网络地址
KV: 服务器 V 与 AS 共享的密钥
P_C: 用户 C 的口令
设有用户 C，服务器 V，通过 AS(识别服务器,与用户共享口令，与其他服务器共享密钥)提供服务。
首先，用户 C 向 AS 提供自己的用户名 IDc、口令 Pc、服务器名 IDv；
接着，AS 为用户返回一张票据 Ticket，它是用 AS 和服务器共享的密钥加密的一条消息，包含 C 的用户名、网络地址以及 V 的名称；
最后，C 通过发送自己的名称和从 AS 处得到的票据来获得访问服务器 V 的权限。
缺点：P_C 不能明文传送，否则不安全；(p324,325)
Ticket 只能用一次，否则易遭重播攻击；

为换得不同的通行票，用户口令要反复使用，很不安全。

(2)一个更安全的认证会话交互过程：



首先需要注意图中将上面方案的 AS 分成了两个部分，识别服务器 AS 和通行票产生服务器 TGS。AS 中存储与用户和 TGS 共享的密钥，专门负责识别用户身份，然后将 TGT(Ticket-Granting Ticket)用与 TGS 共享的密钥加密交给用户。用户据此获得 TGS 的服务，TGS 产生 SGT(Service-Granting Ticket)，用户据此获得其他服务器的服务。

①ID_C, ID_{tgs}, TS_1

②$E_{KC}[K_{C,tgs}, ID_{tgs}, TS_2, Lifetime1, Ticket_{tgs}]$

$Ticket_{tgs} = E_{Ktgs}[K_{C,tgs}, ID_C, AD_C, ID_{tgs}, TS_2, Lifetime2]$

③ID_V, $Ticket_{tgs}$, $Authenticator_C$ (用户身份证明文件)

$Authenticator_C = E_{kc,tgs}[ID_C, AD_C, TS_3]$

④$E_{kc,tgs}[K_{C,V}, ID_V, TS_4, Ticket_V]$

$Ticket_V = E_{kv}[K_{C,V}, ID_C, AD_C, ID_V, TS_4, Lifetime4]$

⑤$Ticket_V$, $Authenticator_C$

$Authenticator_C = E_{kc,v}[ID_C, AD_C, TS_5]$

⑥$E_{kc,v}[TS_5+1]$

①用户向 AS 提供自己的用户名 IDc、TGS 服务器的名称 ID_{TGS}、以及当前的时间戳；

②AS 返回给一个用和用户共享的密钥加密的信息，内容包括用户 C 给 TGS 发送信息用到的密钥 $K_{C,tgs}$ 等，最重要的是 $Ticket_{tgs}$，它是一个用 AS 和 TGS 共享密钥加密的消息(即用户 C 无法查看其内容)，包括 $K_{C,tgs}$ 等信息，这样一旦票据交给 TGS，TGS 就能和用户 C 共享密钥 $K_{C,tgs}$ 了；

③用户 C 向 TGS 发送票据和用户身份证明文件等消息；

④TGS 得到后，即可进行验证，用于 AS 共享的密钥解开票据，得到与 C 共享的密钥，再验证 C 的身份证明文件。验证合法后，给 C 发送用 $K_{C,tgs}$ 加密的信息，内容包括 C 和 V 通信的密钥 $K_{C,V}$ 和票据等信息，票据又是用 TGS 和 V 共享的

密钥加密的消息(即用户 C 无法查看其内容)，它包括 C 和 V 共享的密钥等信息；
⑤C 向服务器 V 提供票据和身份证明文件；
⑥V 返回密钥加密的时间戳+1 信息，表示正常收到该信息，会话建立成功。

这里如果攻击的话就是截获两张票据然后在有效时间内进行**重放攻击**，解决办法是在提交 TGT 或 SGT 的同时必须向服务器证明其身份仍同前面取得 TGT 和 SGT 时的用户相同，**即提交票据时必须同时提交身份证明文件**，其中包含用户名、网络地址和时间戳，这样就能防止重放攻击了。
另外，使用 AS、TGS 和 V 三层结构能够有效地减少用户验证的次数，即①②仅在用户登录时执行一次，③④仅在用户请求一类服务是执行一次，⑤⑥才是用户每申请一次服务执行一次。换句话说，用户如果切换服务类型，则③④仅执行一次即可，①②道理类似，且执行次数更少。
(3)多重 Kerberos 认证服务

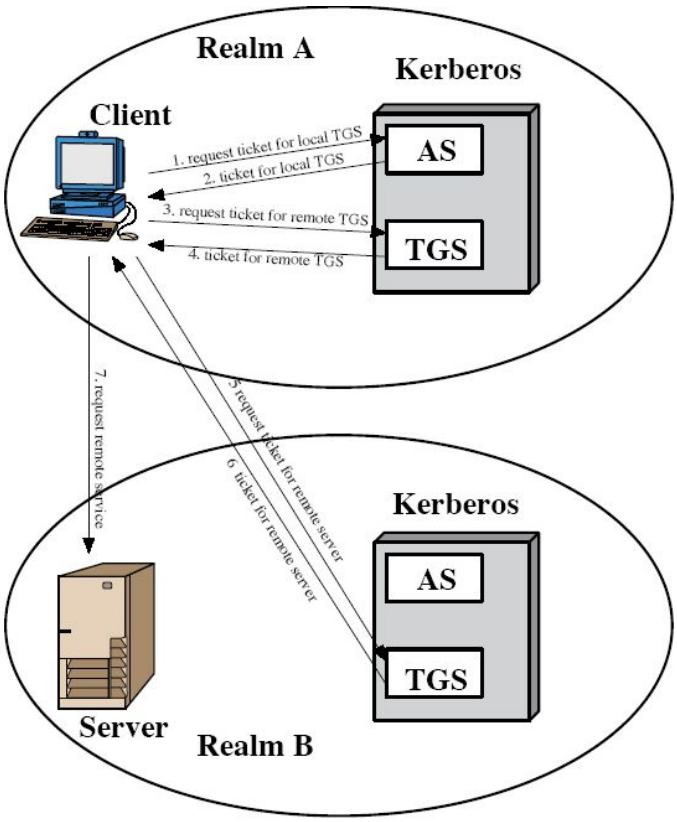
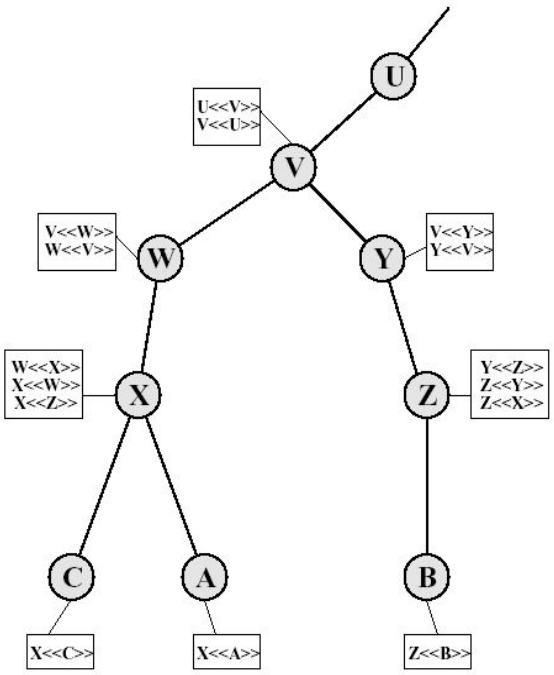


Figure 14.2 Request for Service in Another Realm

这张图描绘了 Kerberos 域间认证的机制，可以看到不同域中的服务器**互相注册**，共享密钥，这样用户 C 就可以通过本域内的服务器访问到域外的服务器了，但是会话建立过程会比域内多两轮，而且是建立 **TGS 服务器**上的。

7. X.509 认证服务分析

X.509 是关于证书结构和认证协议的一种重要标准，它是基于公钥密码体制和数字签名的服务，公钥密码体制使用 RSA，数字签名使用散列函数。
协议最有特点的就是它的证书颁发是树形结构的：



结果中，每一个用户信任他的父节点的证书；
每个 CA 目录入口包含两种证书
向前证书：由其他 CA 发给 X 的证书(forward, client)
向后证书：X 发给其他 CA 的证书(backward, parent)
看着图能把路径写出来就行了，有规律可以找，不理解就死记
A 从目录获得证书以建立通往 B 的证书路径：
X<<W>>W<<V>>V<<Y>>Y<<Z>>Z<<B>>
B 通过如下路径获得 A 的公开密钥：
Z<<Y>>Y<<V>>V<<W>>W<<X>>X<<A>>

结语：留着考完试写……