**数字签名：发送方用自己私钥对消息（或消息散列值）加密就是该用户对该消息的数字签名，可以保证发送方不可否认。**

**数字证书：数字证书是一个经认证中心CA数字签名的包含公钥拥有者信息及其公钥的文件。**

**数字信封：在实际应用中，并不直接使用公开加密算法加密明文，而仅用它保护实际用于加密明文的对称密钥。**

**数字指纹：消息的散列值（或MAC值），即消息摘要。**

**对称加密算法**

**优点：运算简单、易于实现，占用资源少，加密速度快。**

**缺点：1）进行安全通信前要以安全方式进行密钥交换，这在互联网环境下非常困难。**

**2）密钥规模非常大：N个用户两两进行相互通信，对每个用户需要维护N-1个密钥，总共需要(N-1)N/2个密钥。密钥管理困难。**

**公开加密算法**

**1）RSA：大合数分解成大质数的困难（可用于加密和数字签名）**

**2）D-H：离散对数（只用于密钥交换）**

**3）椭圆曲线：公开加密算法的发展方向**

**4）数字签名标准：DSS**

**相互通信的每个用户生成一对密钥（N个人共有2N个密钥），一个可以公开发布（公钥），另一个自己使用，只能自己知道（私钥）。**

**1024位强度才够，理解为什么需要长？**

**1）对称密钥完全保密的，所以短密钥就可以具有高强度；**

**2）公开密钥算法毕竟公钥已公开，公钥和私钥肯定有一定联系（尽管难以推导），所以长密钥才安全。**

**公开密码技术特点**

**优点：1）通信双方事先不需要通过保密信道交换密钥。**

**2）密钥持有量大大减少。N个用户两两相互通信只需要拥有N对密钥（每个用户一对密钥）。**

**3）公开密码技术还提供了对称密码技术无法或很难提供的服务，如与Hash函数联合运行可生成数字签名。**

**缺点：大量的浮点运算致使计算量大，加密/解密速度慢，需占用较多资源，对于电子商务活动尤为突出。**

**数字信封技术**

**在实际应用中，并不直接使用公开加密算法加密明文，而仅用它保护实际用于加密明文的对称密钥，即所谓的数字信封技术。使用这种方式，用户可以在每次发送保密信息时都使用不同的对称密钥，从而增加密码破译的难度。一次密码的破译不影响其他传递。**

**椭圆曲线加密方法与RSA方法相比，有下述优点：**

**安全性能更高，**

**计算量小，处理速度快（密钥短得多），**

**存储空间占用小和带宽要求低等。**

**接收端不可否认——数字签名收条（S/MIME v3）**

**1）A向B发送一封带数字签名的消息，B如果只有A的公钥并不能解密消息。**

**2）解密消息之前，他必须到A处取得一样信物，有了它再加上发送方公钥才能得到消息明文。**

**3）但是他取信物时就在发送方留下日志，事后就不能否认。**

**4）只要他在发送方留下取信物的日志，他就不能否认他收到了消息——日志就相当于数字签名收条。**

**时间不可否认（抗重放）——时间戳**

**量子密码**

**某种程度上是一次一密思想的一个变体**

**——产生和传输数字（随机数）**

**1）为实现量子密码，我们需要随机但可检测方向的光源（发送者使用的光子枪），和很可靠地过滤接收到光子的方法。**

**2）超过20km的可靠通信已经成功 50km**

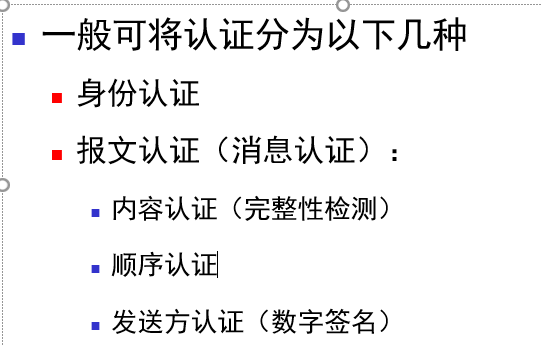
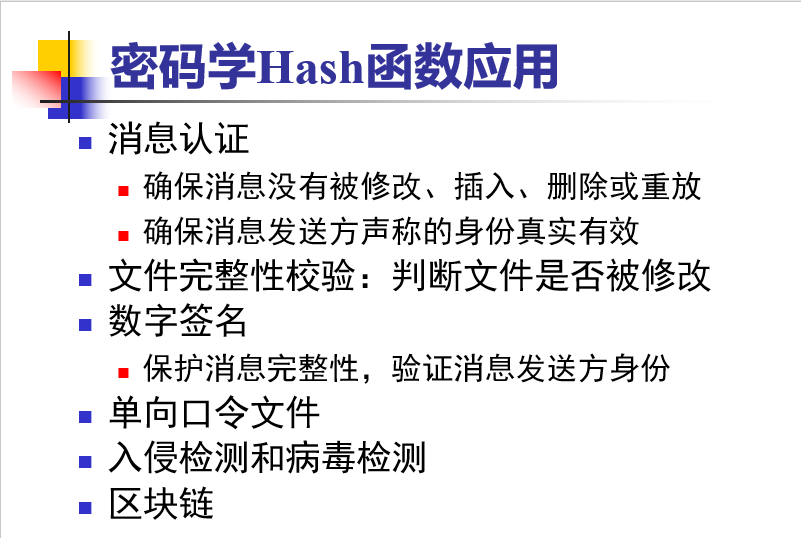
**3）20年商业使用，有可能走在量子计算机的前面。**

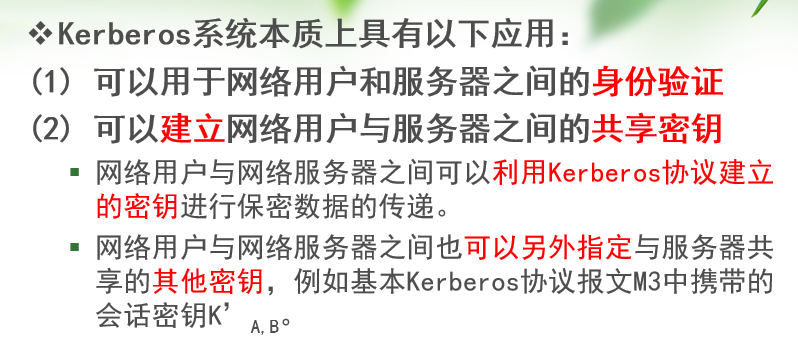
**如果有第三方试图窃听量子密码，就必须用某种方式测量它，而这些测量会带来可察觉的异常，通讯的双方便能察觉。**

**主因是任何对量子系统的测量都会对系统产生干扰。**

**透过量子叠加态或量子纠缠态来传输数据，通信系统便可以检测是否存在窃听。**

* 认证（Authentication，又称为鉴别、确认）是证实信息交换过程合法有效的一种手段，主要解决以下一些问题：
  1. 信息的访问者是不是系统的合法用户？
  2. A和B在进行信息交换时，A和B都必须对对方的身份进行认证，以保证所收到的信息是由确定的实体发送的；
  3. A向B发送信息，B作为接收方，必须确定信息在A发出以后是否被修改过；
  4. 对收到的信息，B必须确定是过时的信息还是某种信息的重放，还是伪造的消息；
  5. 发送方不能否认它所发送过的信息；





* **安全性分析**
  + 一旦用户获得过访问某个服务器的许可证，只要在许可证的有效期内，该服务器就可根据这个许可证对用户进行认证，而无需KDC的再次参与；
  + 实现了客户和服务器间的双向认证；
  + 支持跨域认证；
  + 应用广泛，互操作性好。
  + Kerberos认证中使用的时间戳机制依赖于域内时钟同步，如果时间不同步存在较大的安全风险；
  + Kerberos无法应付口令猜测攻击；
  + 主体必须保证他们的私钥的安全；
  + Kerberos中AS和TGS采用集中式管理，容易形成瓶颈。
* **针对Kerberos系统的不足，必须采取措施：**

**（1）要求在建立Kerberos系统时，一定要严格管理用户注册和口令的录入。**

**（2）要求采用专门服务器作为身份认证服务器。**

**（3）要求对身份认证服务器进行严格的安全保护，防范其中用户口令信息的泄露。**

**典型应用中MAC的访问控制关系分为两种：**

**1) 下读/上写：保证数据机密性**

**2) 上读/下写：保证数据完整性**

**下读：低级别用户只能读比它信任级别更低的敏感信息；**

**上写：只允许将敏感信息写入更高敏感区域。**

**此时信息流只能从低级别流向高级别，保证数据的机密性（Bell-Lapadula模型） 。**

**举例：防火墙所实现的单向访问机制。**

**它不允许敏感数据从内部网络流向Internet：**

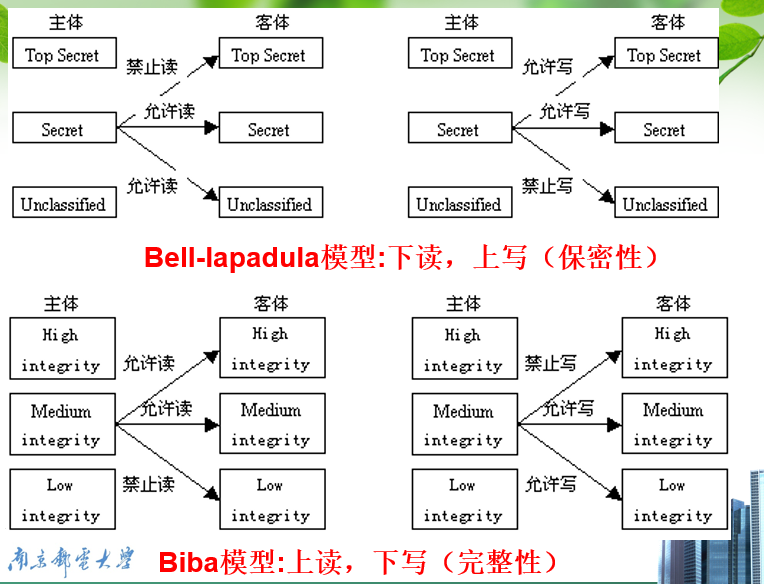
**1）防火墙提供的下读功能来阻止Internet对内部网络的访问，**

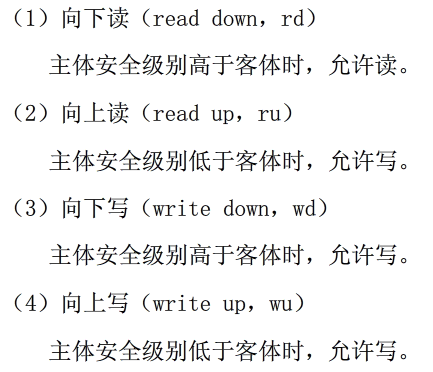
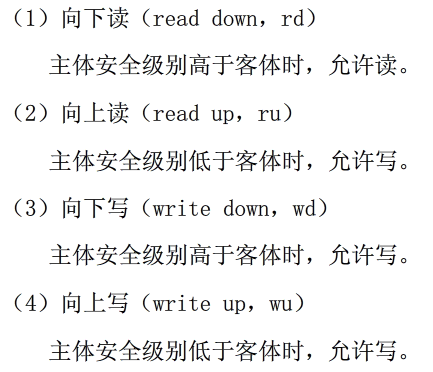
**2）提供的上写功能限制进入防火墙的数据流只能经由由内向外发起的连接流入。**

Biba模型解决了系统内数据的**完整性**问题。它不关心安全级别和机密性。Biba模型用完整性级别来防止数据从任何完整性级别流到较高的完整性级别中。信息在系统中只能自上而下流动，基本安全策略是“**上读下写**”

**对Web服务器的访问过程。**

**将Web服务器发布的资源安全级别看成是秘密，Internet上的用户的安全级别看成是公开。**

 **依照Biba模型，Web服务器上的数据完整性将得到保障，Internet上的用户只能读取服务器上的数据而不能更改。**



**RBAC优点**

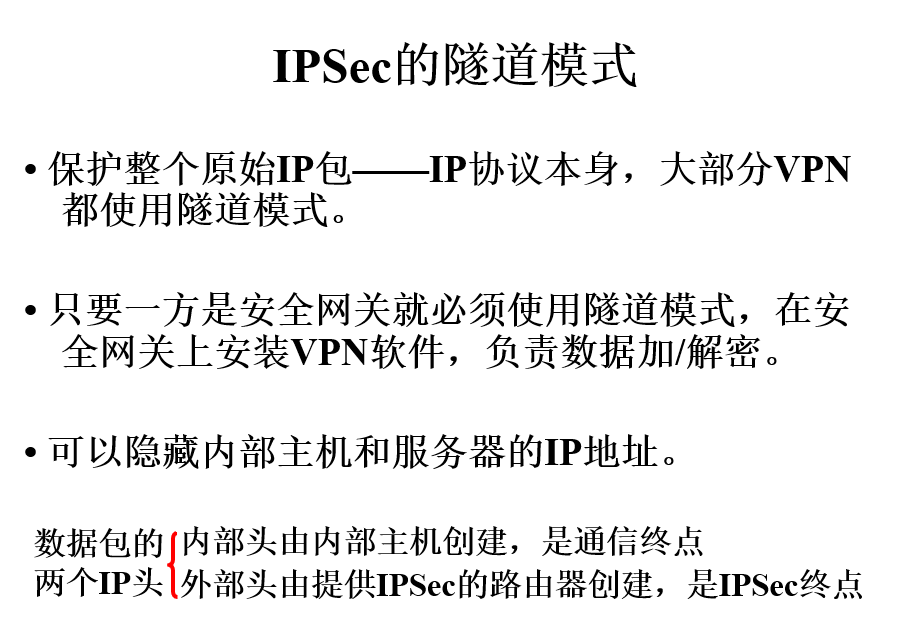
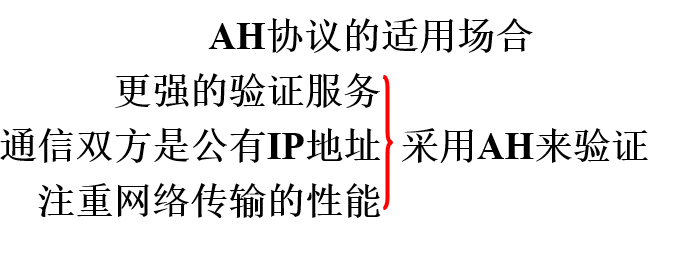
**角色/权限的变化比角色/用户关系之间的变化相对要慢很多。**

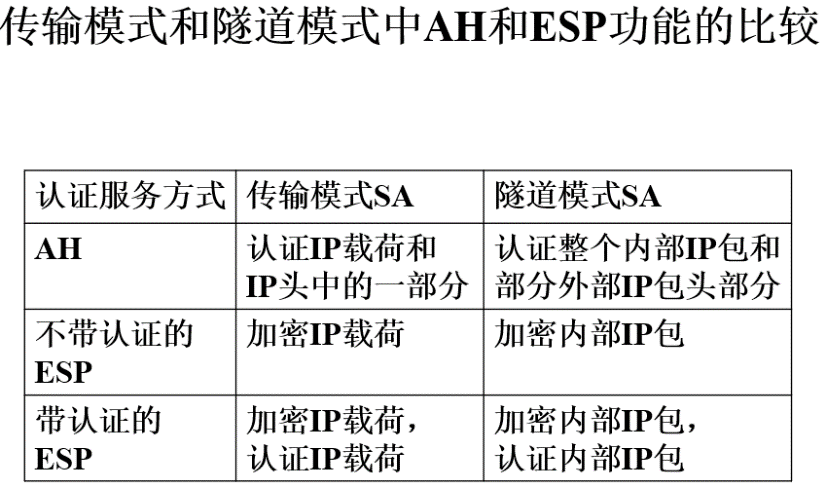
**1）给用户分配角色不需要很多技术——由行政管理人员完成；**

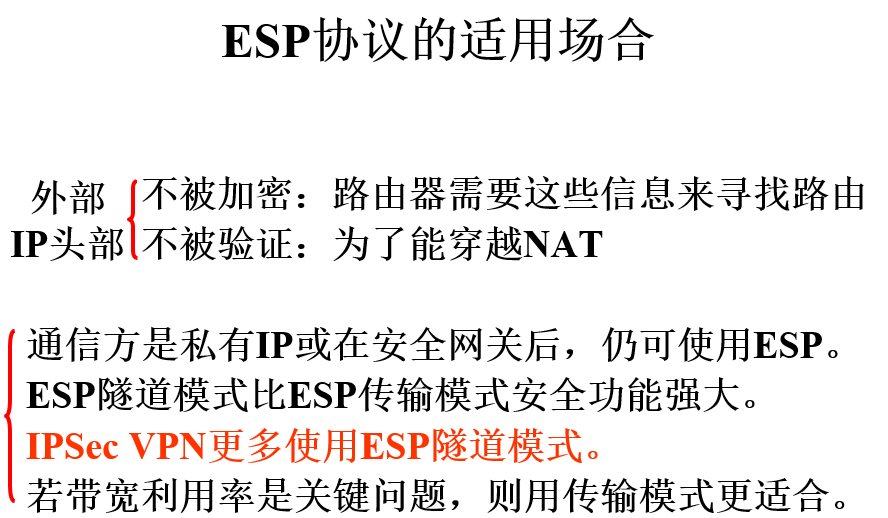
**2）给角色分配权限工作复杂——由专门技术人员承担，但他们没有第一个权限。**

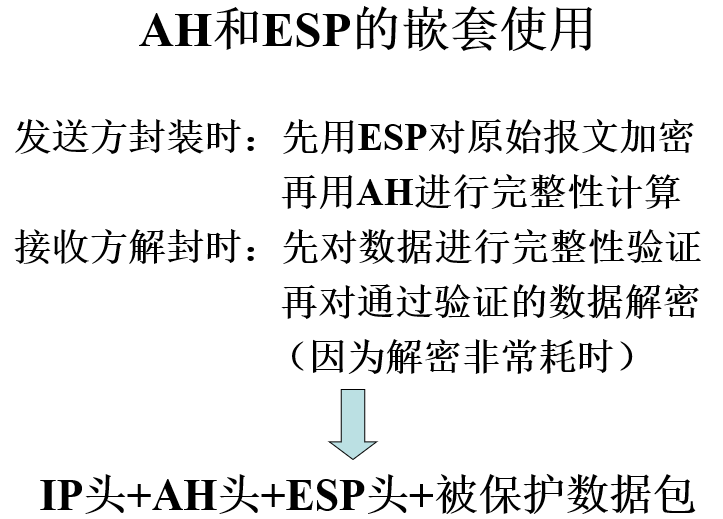
**这两种分配机制与现实中的情况正好一致。**

**最小权限原则和职责分离原则**











**根据隧道建立的网络层次，可分为二层隧道协议、三层隧道协议、应用层隧道协议。主流的二层隧道协议包括PPTP（点对点隧道协议）、L2TP（二层隧道协议），三层隧道协议包括GRE（通用路由封装协议）、IPSec，应用层隧道协议包括SSL/TLS VPN**

* **AH传输模式与隧道模式对比**

**传输模式不改变IP头，隧道模式将整个IP包纳入保护范围，所以需要一个新的IP头。隧道模式要比传输模式消耗更多的资源**

**传输模式适合于端到端之间的安全通信，必须用公网IP地址，因此不能隐藏内网IP地址；而隧道模式适合网关到网关之间或主机与网关之间（即只有一方是网关，则必须用隧道模式），可以隐藏内部主机和服务器的IP地址（因为保护了整个原始IP包）**

* **ESP传输模式与隧道模式对比**
  + 隧道模式对整个IP包进行认证和加密，因此可以提供数据流加密服务，而传输模式由于IP包首部不被加密，因此无法提供数据流加密服务。但是，隧道模式由于增加了一个新IP首部，降低了链路的带宽利用率。
  + 传输模式适合于保护支持ESP协议的主机之间的通信连接，而隧道模式则在包含防火墙或其它用于保护可信内网不受外网攻击的安全网关的配置中比较有效
* **IKE对Diffie-Hellman的改进：**
  + 采用Cookie机制来防止拥塞攻击；
  + 允许双方协商得到一个组（group），相当于Diffie-Hellman密钥交换的全局参数；
  + 使用现时值来阻止重放攻击；
  + 允许交换Diffie-Hellman的公钥值；
  + 对Diffie-Hellman交换进行身份认证，以阻止中间人攻击

PGP

**1）先签名，后压缩**

**若对压缩的消息签名，对验证签名的结果进行动态解压缩的算法则很不稳定**。

**2）先压缩，后加密**

**• 减少了网络传输时间和磁盘空间**

**• 压缩实际上也是一种“混淆”**

**• 先加密后压缩，压缩效果较差**

**链式加密（数字信封）**：

**1）用接收方B的公钥PUB加密对称的会话密钥KS**

**2）以KS为IDEA密钥对压缩的明文加密**，**既有RSA加密的保密——安全传递IDEA密钥**，**又有IDEA算法的快捷——保证消息自身的安全**

**优点：**

**1）发送方随机产生KS，不需和接收方协商**

**2）一次KS的泄漏不影响其他次密文传递的安全**

**3）用RSA公钥PUB对KS加密保证加密的速度**