**1.抵抗重放攻击的方法有哪些？分别应用于哪些场景？各有什么优、缺点？**

**要点：（1）序列号，浪费计算资源但是算法时间复杂度小；（2）时间戳，对于通信双方同步性要求较大，但是流程简单；（3）挑战/应答，要求通信前完成握手，不适用于无连接服务。**

**参考答案：**

对付重放攻击的方法主要有以下几种：

（1）序列号：在每个用于认证的消息后附加一个序列号，只有序列号正确的消息才能被接收。但是这种方法存在一个问题，即它要求参与通信的每一方都必须记住其他通信各方最后附加的序列号。如果身份认证是一个不确定的过程，则通信一方要记住所有可能通信参与方的最后序列号，这是不可行的，特别是对于资源受限的实体更是如此，因此在身份认证过程中，一般不使用序列号。

（2）时间戳：在每个用于认证的消息后附加一个时间戳，接收方在看到这个时间戳后，与其所认为的当前时间进行比对，当两个时间足够接近时，接收方才认为收到的消息是新消息，否则丢弃。但这种方法要求通信各方的时钟保持同步，一旦通信一方的时钟机制出错而使同步失效，则攻击成功的可能性将会增大；同时，在有些环境中实现时钟是很困难的，如：网络延时不可预知的网络环境。

（3）挑战/应答：若 A 要接收 B 发来的消息，则 A 首先要给 B 发送一个临时交互号（挑战码），并要求 B 发来的消息（应答）中包含该临时交互号。但这种方法不适合面向无连接的应用，如：邮件系统。因为它要求在任何无连接传输之前必须先握手，这与无连接信息交互的特征相违背

**2.网上银行可以采用哪些身份认证方法？为什么？分析说明。**

**要点：（1）网上银行认证一定是双向认证；（1）说明两种身份认证技术；（2）分析身份认证方法的特点，并和网上银行的安全性需求相比对，可以讨论现实应用采用了何种原理的身份认证。**

**参考答案：**

用户验证手机银行的方法

（1）网上银行不仅需要验证用户的身份，而且需要验证网上银行的身份。验证网上银行的身份一般有2种：1）手机端的网上银行。由应用商店验证其身份，即将我们验证的权利交给了应用商店，我们子所以相应app，是因为我们相信应用商店的公信力。2）电脑端的网上银行。采用https连接网上银行，通过验证该url地址的数字证书来验证该网站的安全性。

网上银行验证用户的方法：

（1）基于公钥密码技术的身份认证（u盾）。网上银行业务，涉及到登录、支付和转账业务，因此可以采取数字签名技术，以保证用户的消息内容、用户的身份、用户业务操作的时间得到认证。广泛使用的动态口令卡等产品，也是基于数字签名原理，采用挑战/应答的方式来抵抗重放攻击，实现双向认证。数字签名相较于静态的登录密码等能够应对更高的安全需求。

（2）基于生物特征提取的身份认证。面容识别、指纹识别、虹膜识别等技术能够提取出用户独一无二难以伪造和复制的生物特征数据，具备普遍性、唯一性、安全性和可测量性。对于网上银行中二次认证或者一些安全要求较低的业务，基于生物特征提取的身份认证在节省大量流程的同时，具备一定的可信度。

**3.什么是单向认证协议？什么是双向认证协议？**

**要点：（1）仅接收方能够验证发送方身份的为单向认证协议；（2）上方双方相互认证的。**

**参考答案：**

身份认证分为单向认证和双向认证，如果通信过程中只需要一方验证另一方的身份，则这样的认证过程为单向认证，如电子邮件系统；如果通信过程中需要相互认证通信参与方的身份，则这样的认证过程为双向认证，如登录电子银行处理相关业务。

单向认证协议的接收方通过验证发送方或仲裁者的签名结果，来验证发送者的身份。双向认证协议又称为相互认证协议，可以使通信双方在确认对方身份的基础上交换会话密钥。在这个过程中，保密性和及时性是身份认证中的两个重要问题。为了防止用户的身份被假冒，代表用户身份的特定抽象对象必须以密文的形式传送，这时需要验证密文是否为重放消息，在最坏的情况下，攻击者可以成功地假冒通信一方参与整个通信过程。

**4. 对称加密算法能否作为身份认证的依据或方法？为什么？如果希望将对称加密算法应用于身份认证，该怎么做？这样做有什么缺陷？**

**要点：(1)对称加密算法无法唯一标识用户的身份，难以抵抗消息被篡改或者用户被中间人假冒的情况。（2）可以解决第三方，实现基于对称密钥的身份认证。（3）无法抵抗第三方和通信参与方合谋欺骗。**

对称加密算法是通信双方所持有，接收方通过验证发送方是否持有该密钥来验证发送方的身份，一旦出现纠纷，无法进行裁决，这是因为对称密钥无法唯一标识用户的身份。而公钥密码算法不同，私钥只由用户自己保存，在不泄密的情况下，它可以唯一标识用户的身份。因此，无论是在传输过程中验证用户的身份，还是在后续的仲裁，它都可以作为验证用户身份的依据。

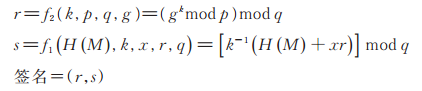
如果希望使用对称加密算法，则可以借助第三方完成身份认证。例如：通信双方分别为A和B，可信第三方为C，A和C共享对称密钥，B和C共享对称密钥。A和B要进行密钥交换，那么首先就要进行相互的身份认证，此时，可借助C来完成。

**5.DSS 对每个签名都会有一个不同的 K，所以即使对同一个消息签名，其签名结果也会不同。这样做有什么意义？分析说明。**

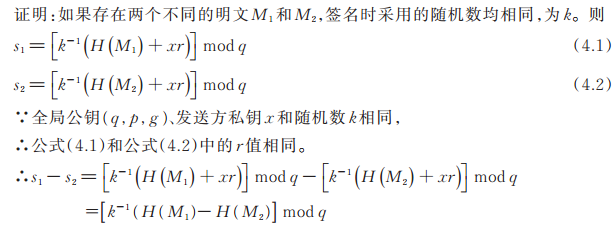
**要点：（1）说明DSA算法原理；（2）证明两次签名一样的情况下可能会暴露用户私钥。**

**参考答案：**

在 DSS 标准中采用的数字签名算法为 DSA。利用 DSA 算法进行签名，需要计算两个函数值 r 和 s，它们是全局公钥，发送方私钥，消息散列值*H(M*)以及随机数或者伪随机数k的函数值。签名的公式计算如下：



当攻击者能够获知任意两次通信双方的消息，并且两次的签名值一样的情况下，能够计算出随机数K，并以此计算出私钥*x*，随后的通信内容都对攻击者透明。该结论的证明如下：



**6.设计一种基于公钥密钥体制实现双向认证的协议。**

**要点：在协议中体现公钥算法；保证双向认证的完成；进行协议的描述和安全性分析。**

**参考答案:**

设计思路：采用公钥体制，对用户身份标识Uid和用户密码Psd用认证服务器的公钥进行加密后，再通过网络通信。同时结合静态加密的口令和基于挑战/应答的动态口令的机制，避免非法用户的重放攻击。

身份认证过程：

（1）在客户端中用户输入Uid和Psd，客户端生成随机数Ru,并用服务器的公钥加密Uid、Psd和Ru，即A=EPks（Uid||Psd||Ru）,发送A到服务器。

（2）服务器收到A，用私钥解密得知上述信息。

（3）服务器在数据库中验证Uid和Psd的有效性。

（4）服务器产生随机数Rs，使用用户公钥加密得到B=Epku(Uid||Psd||Ru+1||Rs)发给客户端。

（5）客户端用私钥解密B后得到上一步的信息，验证Uid、Psd、Ru的有效性。

（6）根据f=（Ru,Rs）计算出对话密钥Ks，使用Ks加密Rs+1为C，将C发给服务器。

（7）服务器解密C后验证，若通过则完成对客户的认证，通知客户，双方以商量好的Ks进行安全通信。

安全分析：该协议密文传输用户名和密码，能够保护用户身份。同时动态口令防止重放攻击。在动态口令认证的过程中加入静态口令认证，解决动态口令暴露系统资源的问题，以实现安全的双向认证。服务器与客户端双方身份不可能被冒充，用户身份安全保密，也能够抗重放和选择密文攻击。

**7. 简述指纹识别、虹膜识别和人脸识别等基于生理特征的身份认证技术，分析它们的优、缺点。**

**要点：（1）说明三种技术采集何种特征，为什么采集这种特征。（2）指纹特征具备唯一性，但是人的指纹由于和外界接触，容易被损坏；虹膜识别精度高于指纹，但是也会受到外界光的影响产生失真；人脸识别简单快捷易被人们接受但是可靠性较差，识别效果不稳定。**

**参考答案：**

指纹识别就是通过比较不同指纹的细节特征点来进行鉴别。指纹识别技术是最早的基于生理特征的身份认证技术。早在我国古代，就利用指纹（手印）画押来识别个人身份，人们通过肉眼进行指纹匹配来识别身份。而在现代，则是通过取像设备读取指纹图像，然后用计算机识别软件分析指纹的特征，从而自动、迅速和准确地鉴别出某个人的身份。

虹膜在胎儿发育阶段形成后，在整个生命过程中保持不变。与指纹不同，一个虹膜约有 266 个量化特征点，在算法和人类眼部特征允许的情况下，算法可获得 173 个二进制自由度的独立特征点，因此，虹膜的误识率为 1/1500000，精确度仅次于 DNA 识别。虹膜识别技术最早使用在 1985 年，当时巴黎监狱仅利用虹膜的结构和颜色区分监狱中的犯人。1993 年，Daugman 提出了最初的虹膜理论框架和算法。在此基础上经过改进后，框架主要包括四个部分：采集，预处理，特征提取和匹配。这个框架与指纹识别框架类似，但获取的是虹膜图像，预处理时需要虹膜定位，以确定内圆、外圆和二次曲线在图像中的位置。其中，内圆为虹膜与瞳孔的边界，外圆为虹膜与巩膜的边界，二次曲线为虹膜与上、下眼皮的边界。其次是归一化处理，这是因为不同人的虹膜大小不同；同一虹膜的大小会随着瞳孔大小的改变而改变，随着外界光照的变化，瞳孔会扩张或收缩；采集图像时，眼睛与采集设备间的距离直接影响瞳孔成像的大小。虹膜的这种弹性变化会影响识别的效果，大小不同的虹膜是无法识别的，因此归一化的目的是纠正这些缩放失真。

人脸识别技术是根据人的面部特征，如眼睛、鼻子、唇部、下颚的形状和尺寸，以及它们之间的相对位置关系来识别个人身份与指纹识别技术类似，人脸识别过程分为四个部分：人脸图像的采集，人脸图像的预处理，人脸图像的特征提取，人脸的匹配与识别。人脸识别技术的特点是直接、方便、友好，易于被人们接受，但缺点是可靠性相对较低，误识率为 2%；脸像会随着年龄变化而变化，而且容易被伪造；一旦脸部整容过，就很难识别。另外，在环境光照发生变化时，识别效果会急剧下降，无法满足实际系统的需要。可能的解决方案有三种：三维图像人脸识别，热成像人脸识别和基于主动近红外图像的多光源人脸识别。前两种技术还远不成熟，识别效果不佳。后一种技术能克服光线变化的影响，在精度、稳定性和速度方面的性能超越三维图像人脸识别，并逐渐走向实用化。

**8.简述步态识别和笔迹识别等基于行为特征的身份识别技术，分析它们的优、缺点。**

**要点：（1）步态行为难以伪造并且易于区分，但是目前提取步态特征的技术尚不成熟；（2）笔迹识别分为脱机和联机两种方式，能够有效识别、特征提取较简单但是识别的难度更大。**

**参考答案：**

步态识别是利用生物（包括人）行走时的方式来识别个体的身份。研究表明，由于人们的肌肉力量、肌腱和骨骼长度、骨骼密度、视觉的灵敏程度、协调能力、经历、体重、重心、肌肉或骨骼受损的程度、生理条件以及个人走路的“风格”等方面存在细微差异，因此，没有完全相同的步态，而且伪造走路的姿势非常困难，几乎是不可能的。与人脸识别类似，步态识别分四个步骤：采集步态视频序列，视频序列预处理，步特征提取，匹配与识别。

尽管步态识别技术是一个非常有前途的身份识别技术，但目前还存在很多的难点，主要表现在行人在行走过程中会受到外在环境和自身因素的影响（如不同行走路面、不同时间、不同视角、不同服饰、不同携带物等因素），导致提取到的步态特征呈现很强的类内变化，其中视角因素是影响识别性能主要的因素之一。当行人行走方向发生变化，或由一个摄像监控区域转入另一个不同位置的摄像监控区域时，都会发生视角变化。

笔迹识别是根据书写者的书写技能和习惯特性，通过笔迹的局部变化识别书写者的

身份。笔迹是书写者的生理特征和后天的学习过程的综合体现，不同的书写者，其笔迹

的差别是比较大的，因此，笔迹识别也是进行身份认证的重要方法之一。计算机笔迹识别根据获取数据的方式不同，分为联机笔迹识别和脱机笔迹识别。联机笔迹识别通过获取笔迹的动态信息，如书写压力、速度、握笔倾斜度等，预处理后提取动态特征，进行匹配后识别书写者身份，因此联机笔迹识别又称为动态笔迹识别，常用的识别模型有改进的二次判决函数、支持向量机和隐马尔可夫模型等。脱机笔迹识别是将笔迹经过扫描或拍照形成二维数字图像，再从图像中提取静态特征进行匹配，从而识别书写者身份 ，因此脱机笔识别又称为静态笔迹识别，常用的方法 有 Gabor算法Gradient算法等。联机笔迹识别的优点是提取的笔迹特征中含有丰富的动态信息，不易模仿和伪造，因此比脱机笔迹识别更准确，但实现起来难度更大。

**9.PKI/CA 在身份认证中的作用是什么？分析其工作原理。**

**要点：（1）产生的数字证书实现公钥与身份的绑定。（2）提供一个具有公信力的可信第三方.（3）阐述其工作原理**

**参考答案：**

PKI基础设施采用证书来管理公钥，通过第三方的可信任机构—认证中心（CA），把用户的公钥和用户的其他身份标识信息绑定在一起，在Internet互联网上验证用户的身份。PKI基础设施把公钥密码和对称密码结合起来，在Internet网上实现密钥的自动管理，以保证网上数据的安全传输。

为了保证网络数字信息的传输安全，除加密算法等措施，还应当建立数字证书作为信任和信任验证措施。CA认证中心在身份认证负责数字证书的发放和管理，这一机构具备权威性、公正性、可信赖性。它主要负责产生、分配并管理所有产于网络通信的实体所需的身份认证信息，每一份数字证书都与上一级的数字签名证书相关联，最终通过安全链追溯到一个已知的第三方可信机构（根CA）。

CA的工作核心就是发放和管理数字证书，在PKI基础上完成身份信息与公钥的绑定。其具体工作描述如下：（1）接受验证最终用户数字证书的申请；（2）证书的审批；（3）证书的发放；（4）证书的更新；（5）最终用户数字证书的查询、撤销；（6）完成数字证书、密钥和历史数据的归档。

**10.查阅资料，简述 Kerberos 身份认证的认证过程（图示说明）。**

**要点：(1)是基于对称密钥的身份认证。（2）协议依赖票据的认证过程，图解应当说明各步骤数据内容以及异步顺序。**

**参考答案：**

Kerberos 是一种网络认证协议，其设计目标是通过密钥系统为客户机 / 服务器应用程序提供强大的认证服务。该认证过程的实现不依赖于主机操作系统的认证，无需基于主机地址的信任，不要求网络上所有主机的物理安全，并假定网络上传送的数据包可以被任意地读取、修改和插入数据。在以上情况下， Kerberos 作为一种可信任的第三方认证服务，是通过传统的密码技术（如：共享密钥）执行认证服务的。

协议的安全主要依赖于参加者对时间的松散同步和短周期的叫做Kerberos票据的认证声明。 下面是对这个协议的一个简化描述，将使用以下缩写：

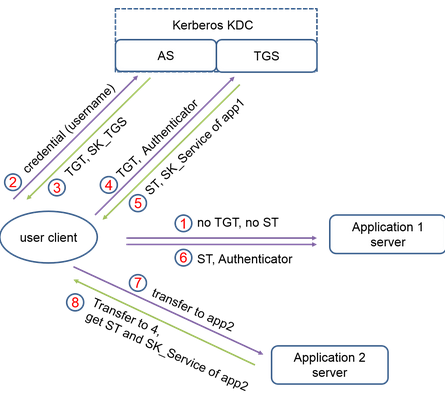
AS（Authentication Server）= 认证服务器

KDC（Key Distribution Center）= 密钥分发中心

TGT（Ticket Granting Ticket）= 票据授权票据，票据的票据

TGS（Ticket Granting Server）= 票据授权服务器

SS（Service Server）= 特定服务提供端



客户端用户发送自己的用户名到KDC服务器以向AS服务进行认证。KDC服务器会生成相应的TGT票据，打上时间戳，在本地数据库中查找该用户的密码，并用该密码对TGT进行加密，将结果发还给客户端用户。该操作仅在用户登录或者kinit申请的时候进行。 客户端收到该信息，并使用自己的密码进行解密之后，就能得到TGT票据了。这个TGT会在一段时间之后失效，也有一些程序(session manager)能在用户登陆期间进行自动更新。 当客户端用户需要使用一些特定服务(Kerberos术语中用"principal"表示)的时候，该客户端就发送TGT到KDC服务器中的TGS服务。当该用户的TGT验证通过并且其有权访问所申请的服务时，TGS服务会生成一个该服务所对应的ticket和session key，并发还给客户端。客户端将服务请求与该ticket一并发送给相应的服务端即可。

简单地说，用户先用共享密钥从某认证服务器得到一个身份证明。随后，用户使用这个身份证明与SS通信，而不使用共享密钥。上述认证交换过程需要只读方式访问 Kerberos 数据库。但有时，数据库中的记录必须进行修改，如添加新的规则或改变规则密钥时。修改过程通过客户机和第三方 Kerberos 服务器（Kerberos 管理器 KADM）间的协议完成。另外也有一种协议用于维护多份 Kerberos 数据库的拷贝，这可以认为是执行过程中的细节问题，并且会不断改变以适应各种不同数据库技术。