1. 了解X.509、数字证书，CRL基本原理和概念

答：

1.X.509标准是ITU-T设计的PKI标准，他是为了解决X.500目录中的身份鉴别和访问控制问题设计的。

2. 数字证书

数字证书的意义在于回答公钥属于谁的问题，以帮助用户安全地获得对方的公开密钥。证书中应对公钥和公钥私有者信息，并由可信任的CA签署，即CA对这些信息进行数字签名。一张数字证书由证书内容、签名算法和算法结果组成。

数字证书的结构如下：

版本号 version

序列号 serialNumber

签名算法 signature

有效日期 vaildity

主体 subject

主体公钥信息 subjectPublicKeyInfo

颁发者唯一标识符 issuerUniqueID

主体唯一标识符 subjectUniqueID

扩展 extensions

3. CRL基本原理和概念

发布证书撤销信息的最基本思路就是：PKI系统中的CA机构将当前被撤销证书的标识（通常是证书序列号）集中到一个列表中，向PKI系统的所有用户公布。和签发证书一样，为了防止伪造和篡改，CA需要对这个列表进行数字签名。

* 使用CRL验证证书的有效性。验证CRL签名上的数字签名是否正确、当前是否处于有效期。
* 构造被撤销证书的证书序列号列表。对于不同的CRL机制，有着不同的方法来构造被撤销证书的信息列表。对于基本的CRL机制，从每一个单独的CRL文件就能够得到完整的、当前被撤销证书的证书序列号列表。
* 检查证书是否已经被撤销。检查被撤销证书的证书序列号列表，查看验证的证书序列号是否在其中。

4. 数字签名

数字签名就是非对称密钥和数字摘要技术的应用。

数字签名就是附加在数字单元上的一些数据，或者是数据单元做得密码变换。这种数据或变换允许数据单元的接受者用以确认数据单元的来源和数据单元的完整性并保护数据，防止被人进行伪造。

发送方用特殊的hash算法，由明文中产生固定长度的【摘要】，然后利用自己的私钥对形成的摘要进行加密，这里加密后的数据就是数字签名。

验证：接受方利用发送方的公钥解密被加密的摘要得到结果A，然后对明文也进行hash操作产生摘要B.最后,把A和B作比较。此方式既可以保证发送方的身份正确性，又可以保证数据在传输过程中不会被篡改。

5.数字信封加密解密原理

数字信封则采用密码技术保证了只有规定的接收人才能阅读信息的内容。

数字信封中采用了单钥加密体制和公钥密码体制。信息发送者首先利用随机产生的【对称密码】加密信息(因为非对称加密技术的速度比较慢)，再利用接收方的【公钥】加密对称密码，被公钥加密后的对称密钥被称之为数字信封。在传递信息时，信息接收方要解密信息时，必须先用自己的私钥解密数字信封，得到对称密码，才能利用对称密码解密所得到的信息。

数字信封既发挥了对称加密算法速度快、安全性好的优点，又发挥了非对称加密算法密钥管理方便的优点。

6. 非对称算法，对称算法，摘要算法，填充算法: RSA/ECC/SM2、DES(CBC/ECB)/3DES/AES、MD5/SHA1/SHA-256,PKCS7Padding。

Pkcs#7是密钥消息语法标准

填充字符串由一个字节序列组成，每个字节填充该字节序列的长度。

数据： FF FF FF FF FF FF FF FF FF

PKCS7 填充： FF FF FF FF FF FF FF FF FF 07 07 07 07 07 07 07

7. 了解ASN.1编码规则：BER、DER

8. 国际标准PKCS系列(如PKCS#7,PKCS#10,PKCS#12)

PKCS#7

signedData

SignedData ::= SEQUENCE {

version Version,

digestAlgorithms DigestAlgorithmIdentifiers,

contentInfo ContentInfo,

certificates

[0] IMPLICIT ExtendedCertificatesAndCertificates

OPTIONAL,

crls

[1] IMPLICIT CertificateRevocationLists OPTIONAL,

signerInfos SignerInfos }

signerinfo type

SignerInfo ::= SEQUENCE {

version Version,

issuerAndSerialNumber IssuerAndSerialNumber,

digestAlgorithm DigestAlgorithmIdentifier,

authenticatedAttributes

[0] IMPLICIT Attributes OPTIONAL,

digestEncryptionAlgorithm

DigestEncryptionAlgorithmIdentifier,

encryptedDigest EncryptedDigest,

unauthenticatedAttributes

[1] IMPLICIT Attributes OPTIONAL }

EnvelopedData type

EnvelopedData ::= SEQUENCE {

version Version,

recipientInfos RecipientInfos,

encryptedContentInfo EncryptedContentInfo }

RecipientInfos ::= SET OF RecipientInfo

EncryptedContentInfo ::= SEQUENCE {

contentType ContentType,

contentEncryptionAlgorithm

ContentEncryptionAlgorithmIdentifier,

encryptedContent

[0] IMPLICIT EncryptedContent OPTIONAL }

PKCS#10

证书请求语法标准。PKCS#10定义了证书请求的语法。证书请求包含了一个唯一识别名、公钥和可选的一组属性，它们一起被请求证书的实体签名（证书管理协议中的PKIX证书请求消息就是一个PKCS#10）。

CertificationRequest ::= SEQUENCE {

certificationRequestInfo CertificationRequestInfo,

signatureAlgorithm AlgorithmIdentifier{{ SignatureAlgorithms }},

signature BIT STRING

}

AlgorithmIdentifier {ALGORITHM:IOSet } ::= SEQUENCE {

algorithm ALGORITHM.&id({IOSet}),

parameters ALGORITHM.&Type({IOSet}{@algorithm}) OPTIONAL

}

SignatureAlgorithms ALGORITHM ::= {

... -- add any locally defined algorithms here -- }

证书发放流程：

1. 用户在RA 注册信息，RA审核通过后，签发证书请求。
2. RA把用户信息传到CA，CA从KMC中取密钥对（一般密钥对由加密机生成）
3. CA把用户信息和公约制成用户证书，并对证书签名。CA把自己的用户证书和用户的私钥通过SSL通路传递给RA。
4. 用户从RA下载证书。

CRL结构：

CertificateList:: = Sequence{

tbaCertList TBSCertList,

signatureAlgorithm AlgorithmIdentifier,

signatureValue BIT STRING  
}

其中tbaCertList又分为：

TBSCertList::=SEQUENCE{  
 signature AlgorithmIdentifier,

Issuer Name,

thisUpdate Time,

nextUpdate Time OPTIONAL

revokedCertificates SEQUENCE OF SEQUENCE{

userCertificate CertificateSerialNumber,

revocationDate Time,

crlEntryExtions Extensions optional  
}

crlExtensions [0]EXPLICIT Extionsions OPTIONALS

}

其中revocationCertificates是CRL最有意义的一部分，对于每一个被撤销的证书，CRL中给出证书序列号，和证书撤销时间。证书撤销时间一定早于CRL更新时间。

常见的证书拓展：

参见RFC3280,常见的拓展有证书策略，证书用途，是否为CA.

公钥基础设施的整体架构。

主要是证书认证中心，证书持有者，依赖方。

辅助组件还包括注册机构RA,资料库系统、密钥管理系统KM，OCSP服务器。