第4章 公开密钥基础设施

南京大学计算机系黄皓教授

2011年10月24日

参考文献

- Warwick Ford, Michael S. Baum著， 劳帼龄译，安全电子商务—为数字签名和加密构造基础设施，人民邮电出版社，2002年5月。
- ISO/IEC 9594-8 Information technology – Open Systems interconnection – The Directory: Public-key and attribute certificate frameworks, Fourth edition 2001-08-01
- ITU-T X.509, Information technology – Open Systems interconnection – The Directory: Public-key and attribute certificate frameworks, 06/97

内容

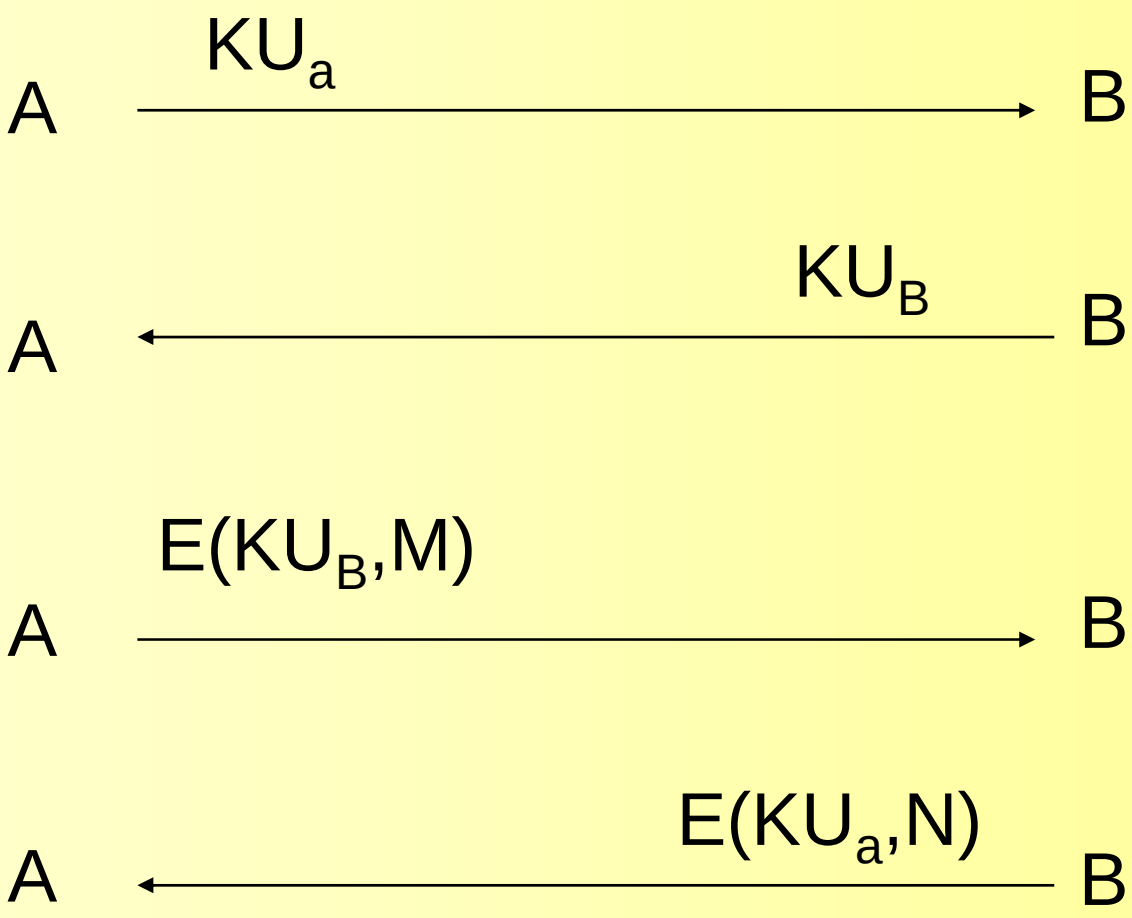
1. 为什么需要公钥基础设施
2. 证书的基本概念
3. X.509 证书格式

1. 信任模型

1. 为什么需要公开密钥基础设施

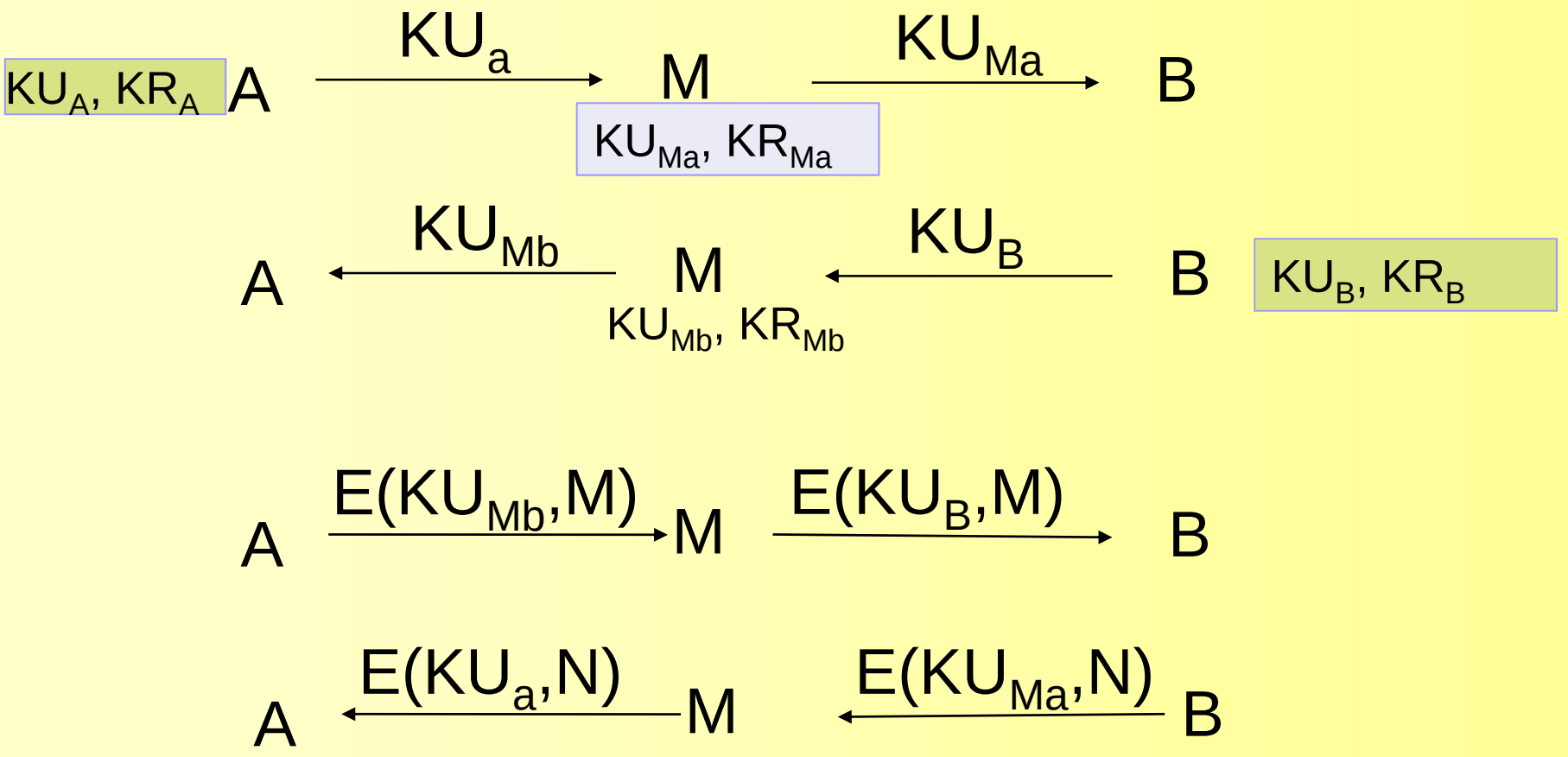
- 中间人攻击
- 抵赖
- 信任

1.1 中间人攻击



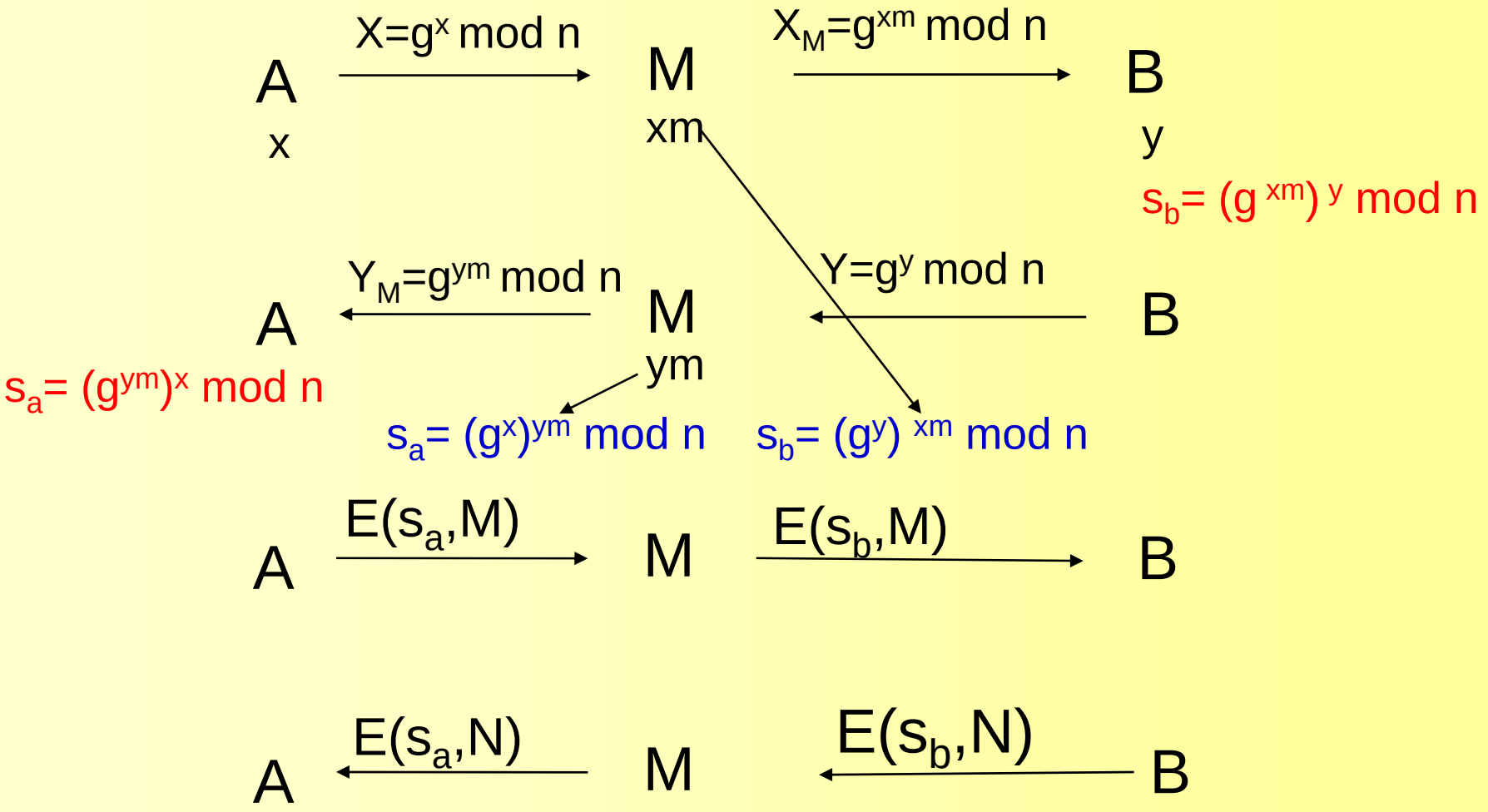
用公开密钥的方式交换密钥

1.1 中间人攻击



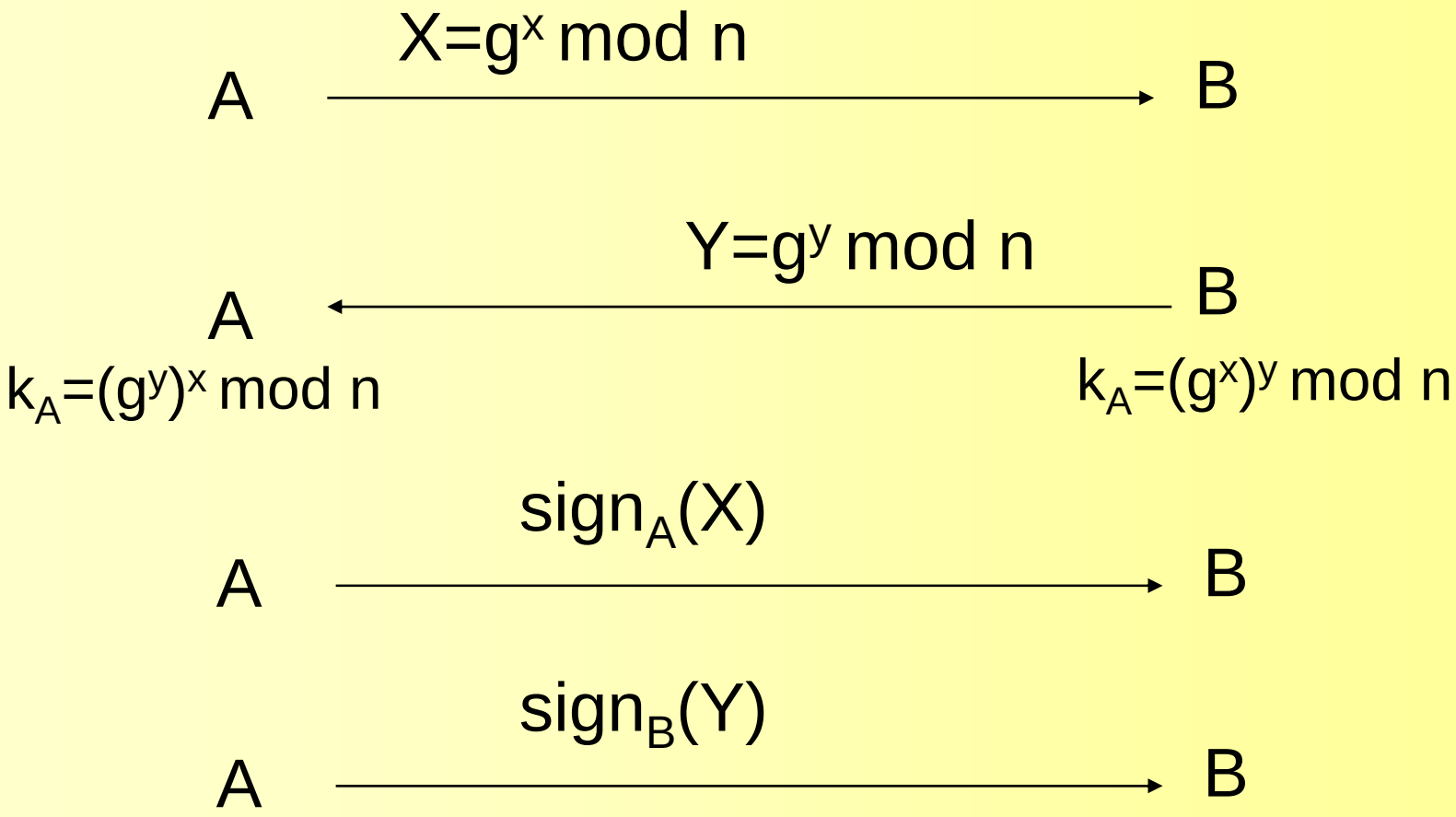
对用公开密钥的方式交换密钥的中间人 攻击

1.1 中间人攻击



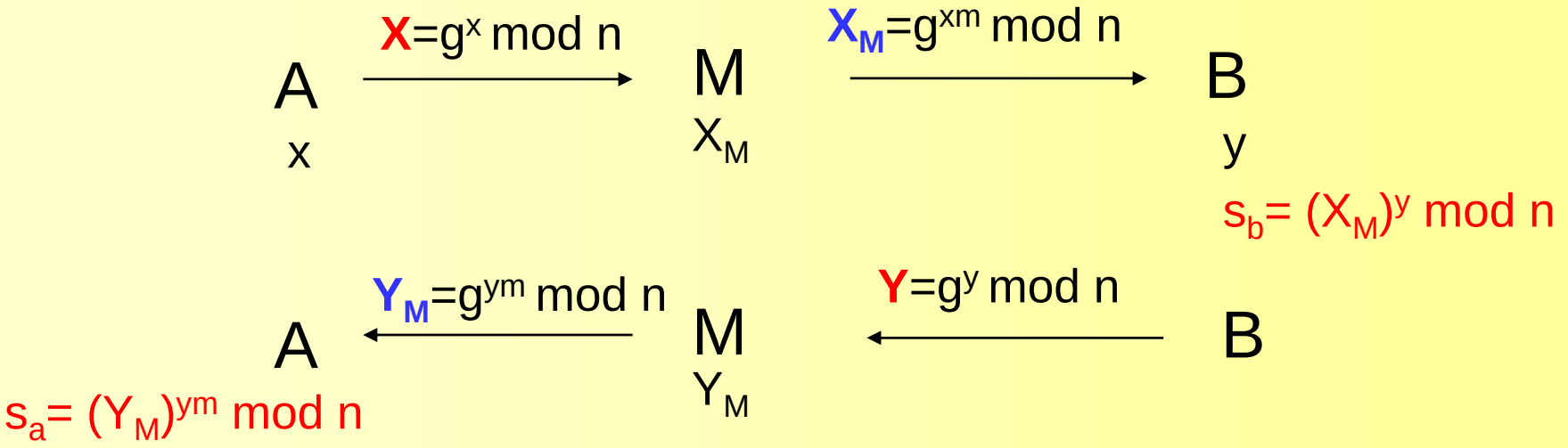
对DH交换密钥的中间人 攻击

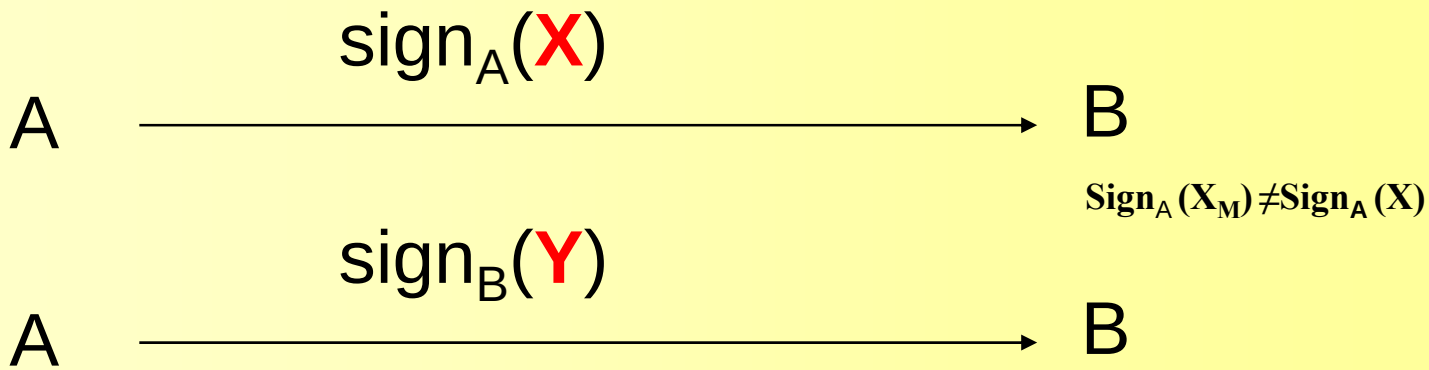
1.1 中间人攻击



对DH交换密钥的中间人 攻击的防范

1.1 中间人攻击





验证公开数时就可以发现是否被篡改了

1.2 抵赖

- A当面将自己的公钥交给B；
- A用自己的私钥签署一份合同交给B；
- B用A的公钥验证合同签名的有效性；
- B按照合同活动；

- A拒绝履行合同中的承诺
  - 声称他没有签署过合同；
    - 那把公钥不是自己的
    - 那把私钥合同之前就丢了，是别人冒充他签的。
  - B用A的公钥验证合同的签名；

- B？？？？？？？

1.3 信任

- 情景
  - B 在一个网站上下载了A的公钥；
  - A用自己的私钥签署一份合同：采购B机器，并把合同在网上发送给B；
  - B用A的公钥验证合同签名的有效性；
  - B交付机器；
  - B向A索要货款，但B收不到回音，也不知道A住在什么地方；

- B思考：
  - 网站和A在合伙欺骗自己吗？
  - A的私钥是在签署合同之前丢失的吗？
  - 网站会在A丢失私钥之后就立即负责任地告诉我吗？如果网站没有及时通知我，网站会负责吗？
  - 如果找到了A，法院能根据签名合同判决吗？
  - 网站的工作人员能准确判断申请者的身份而不被欺骗吗?
  - 如果网站也是被欺骗的，谁对找不到A而对B造成损失负责?

1.4 PKI提供的服务

- 好密钥的安全生成
- 初始身份的确认
- 证书的颁发、更新与终止
- 证书有效性的检查
- 证书和相关信息的分发
- 密钥的安全存档和恢复
- 签名和时间戳的产生
- 信任关系的建立和管理
- ……

1.5 公钥基础设施的组件

- 认证机构
- 注册机构
- 证书服务器
- 证书库
- 证书验证
- 密钥服务服务器
- 时间服务器
- 签名服务器

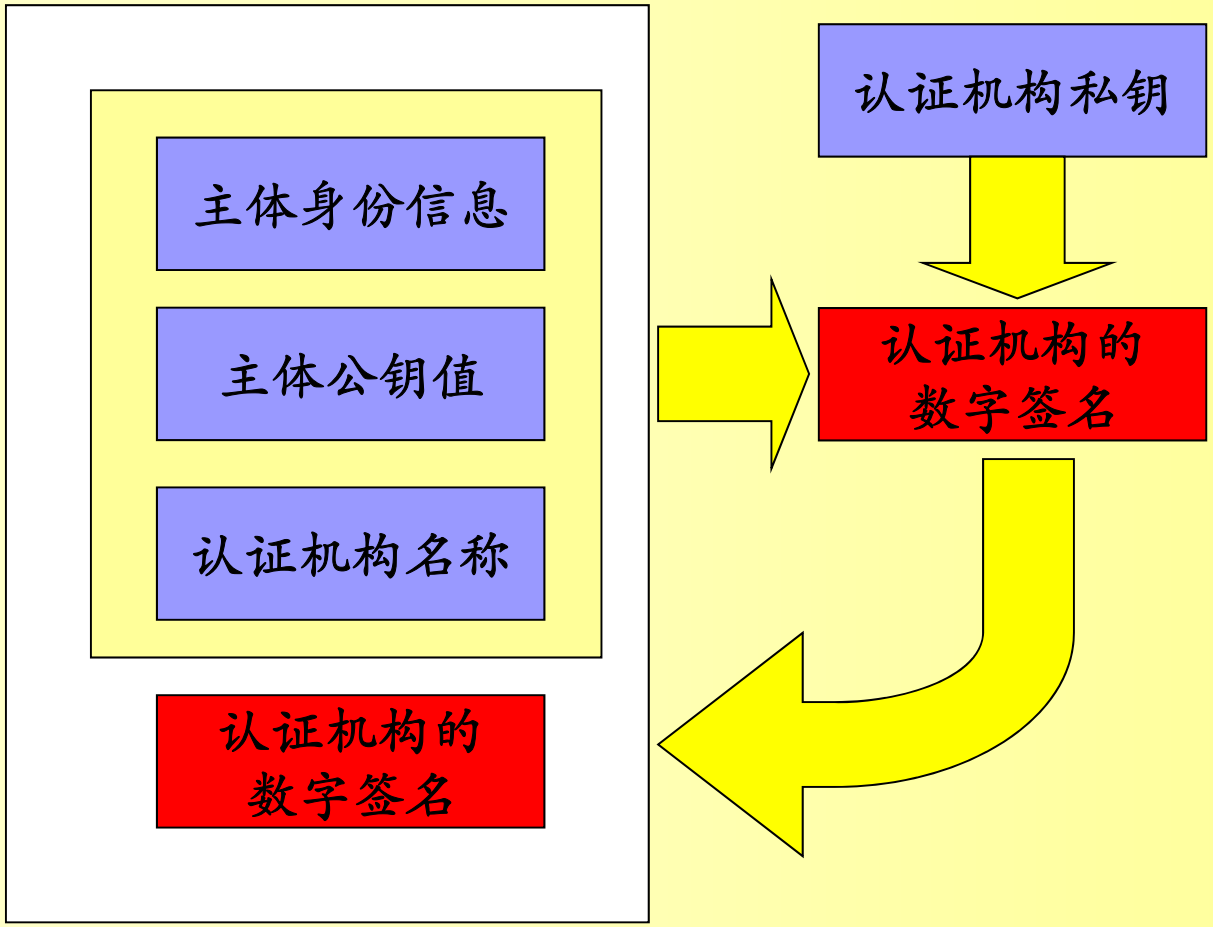
2. 证书的基本概念

2.1 证书的概念

公钥证书

- 公钥证书
  - 证书是一个由使用证书的用户群所公认和信任的权威机构签署的包括证书主体的公钥在内的信息集合。
  - 在该类证书中，一个公钥值与一个特定的实体（人、角色、设备或其他）权威地联系在一起；
  - 公钥证书是由称为证书权威机构（认证机构）的人或实体在确认了相应的私钥持有者的身份或其他属性；
  - 权威机构承诺公钥的主体是证书中公钥配对的私钥的唯一持者。
  - 公开密钥技术和数字签名是电子商务安全的基础
  - 公钥证书则是将这些技术广泛地应用于大型的、全球性的电子商务社区的关键。

公钥证书的结构



公钥证书的意义

- 公钥证书的使用功能
  - 消息发送
    - 当某一消息发送方希望使用基于公开密钥技术的加密方法来发送加密信息时，该消息的发送方必须拥有每一个接收方的公钥拷贝。
    - 如果用的不是接受方的公钥(可能攻击者替换了)，则会造成信息泄漏。
  - 验证数字签名
    - 任一方想要验证另一方的数字签名时，验证方也需要拥有签名方的一个公钥拷贝。
    - 如果用的不是真正的声称的签名者的公钥，则会发生欺骗事件。
    - 如果不能证实数字签名发生时刻，私钥在证书的主体手中，则公钥的主体可以不承认做了签名，可能是事实，也可能是抵赖。

1. 消息的发送者必须知道
   - 公钥没有被第三方替换
   - 私钥没有泄漏

证书机构提供
验证证书主体的服务

2. 数字签名的验证者必须确认证书的主体（对方，即协议的参加方）不能否认签名的发生时刻，私钥在他手中。
   - 密钥对是对方申请的
   - 私钥仍然在对方手中（没有泄漏）

证书机构提供
证书撤销、档案等服务

证书机构提供的是特种服务，资质需要严格管理

公钥证书系统的三类实体

- 认证机构
  - 为公-私密钥对的持有者发放证书。
  - 提供各种服务。
- 证书主体
  - 持有相应私钥的个人、设备或其他实体。
  - 当证书的主体是个人或法人实体时，一般将这类持有证书的主体称为证书机构用户。
- 证书用户
  - 利用证书验证签名；
  - 利用证书发送机密消息。

证书的使用

- 证书用户(公钥用户)
  - 获取证书中的公钥，按照指定的用途使用该公钥的用户；
- 证书用户信任证书权威机构
  - 证书用户相信证书权威机构的承诺，根据合同操作，如果遭遇损失可以得到赔偿。
- 证书用户要验证证书的真伪
  - 证书用户用证书权威机构的公钥验证验证获得的证书是否是他信任的证书机构发放的；
  - 证书用户验证证书是否过期了；
- 证书用户必需安全地获得证书权威机构的签名公钥（或公钥证书）。

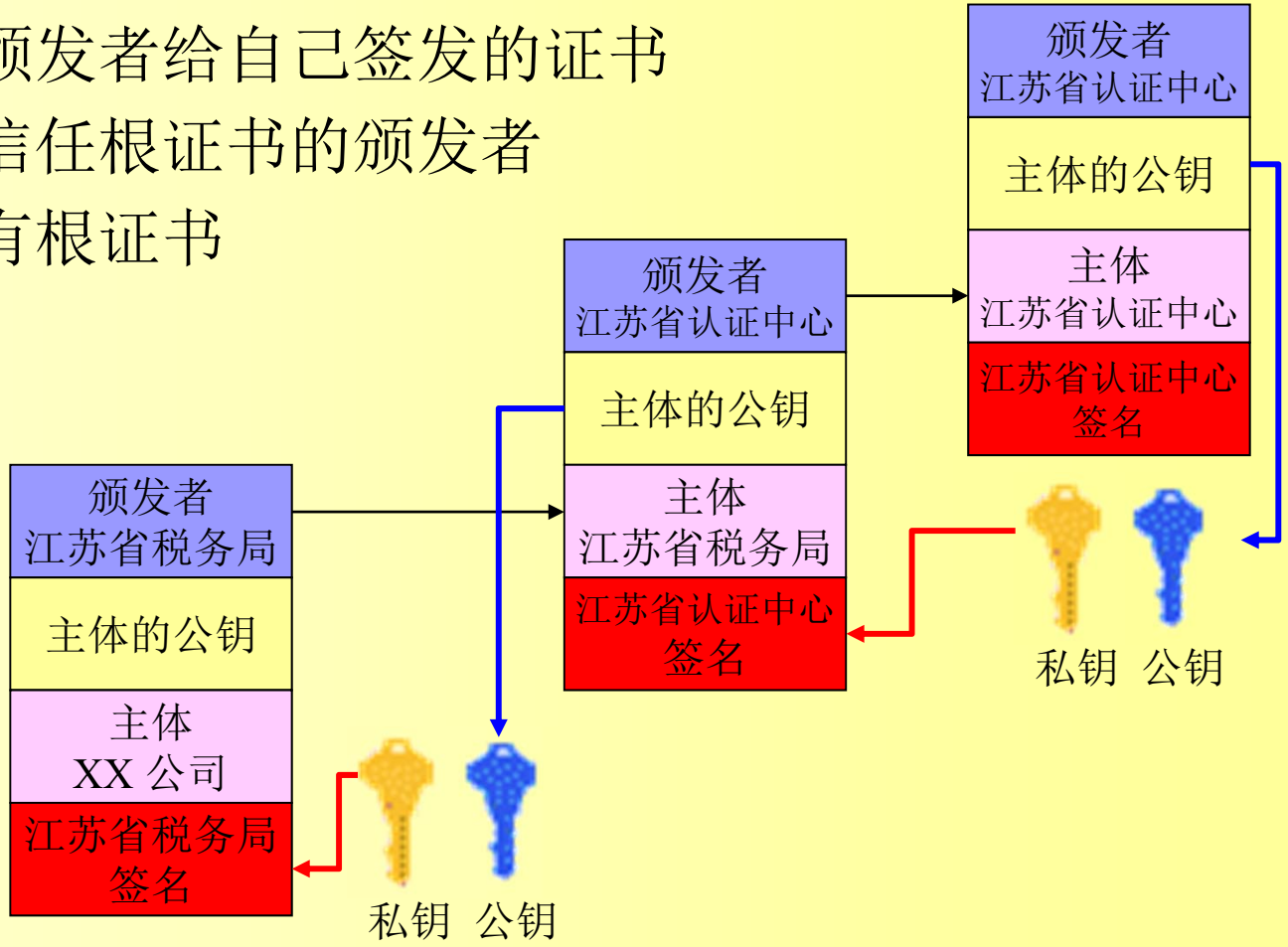
- 公钥证书可以公开的的文件服务器、目录服务系统以及未提供安全性保护的通信协议来分发。
- 使用证书好处主要在于，公钥用户只要知道认证机构的公钥，就可以获得其他很多机构用户的公钥。因而可以获得很好的规模效应，也就是说，只要能运用公钥技术，就可以推广公钥证书的运用。
- 要注意的是，只有当公钥用户相信认证机构的目的仅仅是发放证书时，证书才是有用。

证书的验证

- 证书的主体
- 证书是否在有效期内
- 证书的签名是否正确
- 证书当前是否可用（是否被撤销）
- 证书的用途是否正确

证书的认证 — 认证路径

- 根证书：颁发者给自己签发的证书
- 证书用户信任根证书的颁发者
- 证书用户有根证书



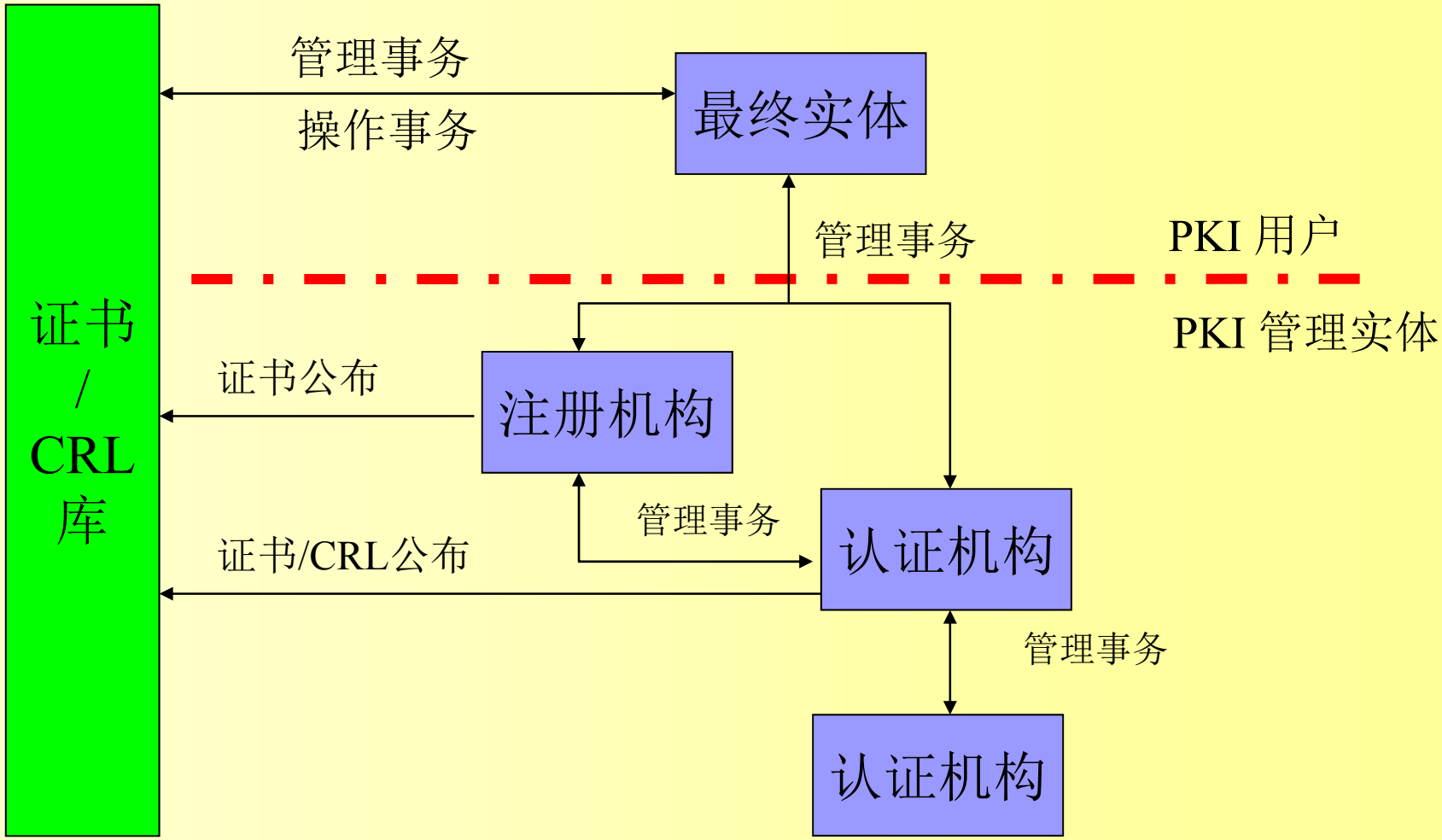
证书 — 法律关系

- 服务对象
  - 专门为某个杜团的内部人员或其附属人员提供服务，如一个组织的内部认证机构可能只管理该组织中雇员的证书；
  - 认证机构为包括非附属人员在内的广大社区提供服务；
- 法律关系
  - 纸上签名与一个特定的人具有内在的联系，因为它们留有签名者的唯一笔迹；
  - 一个用于数字签名的密钥对不与任何人具有内在的联系；
  - 这种联系必须由认证机构确定了持有特定密钥对的人的身份后建立；
  - 利用私钥创建的每一个数字签名的可靠性部分地依赖于将该私钥与签名者相联系的认证机构的可靠性;
- 原则
  - 只有证书主体才能控制相应的私钥；
  - 证书主体必须经过适当的验证

2.2 PKI体系结构 — PKIX

- Public Key Infrastructure X.509 (PKIX)由IETF组成，用来规定证书概要文件集合和操作模型，其适合在Internet上部署X.509公开密钥。

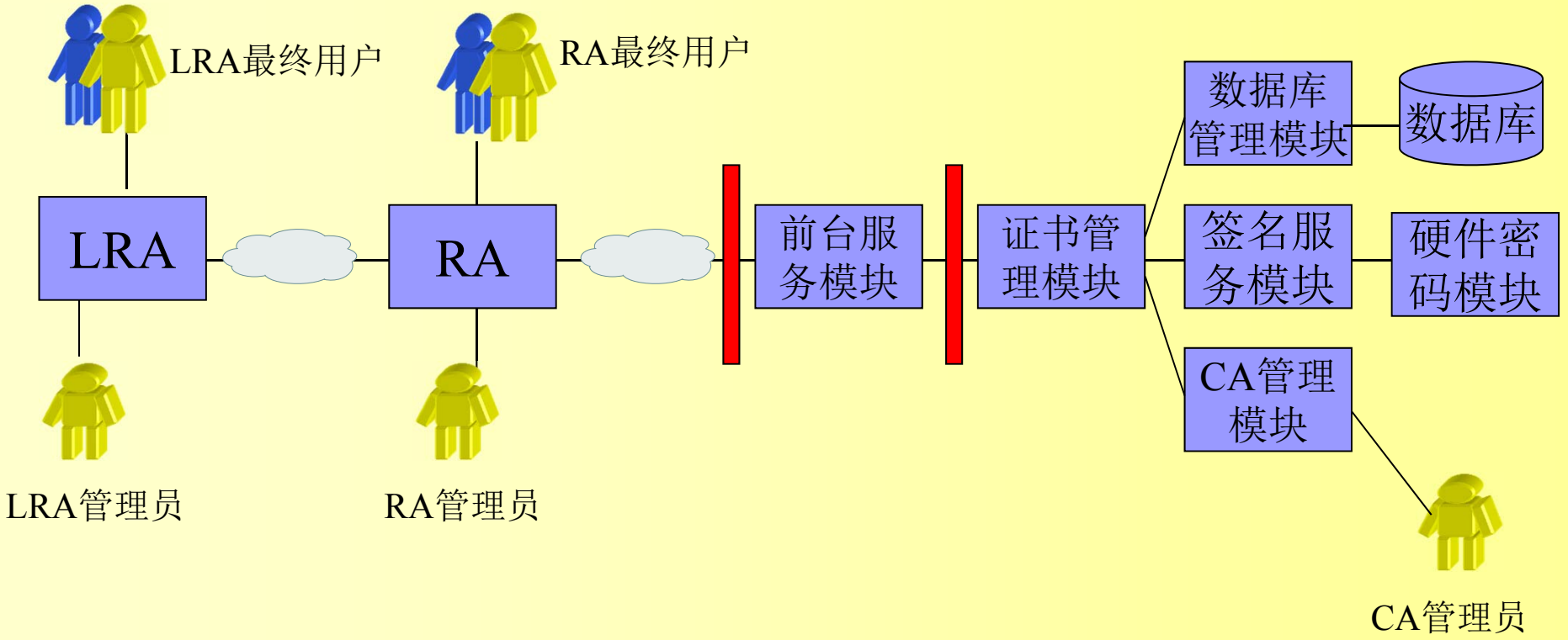
2.2 PKIX 模型


管理事务
操作事务
最终实体
管理事务
PKI 用户
PKI 管理实体
证书
/
CRL
库
证书公布
注册机构
管理事务
认证机构
证书/CRL公布
管理事务
认证机构

2.2 PKIX的功能

- 注册
- 初始化
- 认证
- 密钥对恢复
- 密钥产生
- 密钥更新
- 交叉证书
- 撤销
- 证书与撤销通知分发与发布

CA的网络模型


LRA最终用户
RA最终用户
LRA
RA
前台服务模块
证书管理模块
数据库管理模块
数据库
签名服务模块
硬件密码模块
CA管理模块
LRA管理员
RA管理员
CA管理员

注册机构RA

- 注册机构RA是PKI信任体系的重要组成部分，是用户(可以是个人或团体)和认证中心CA之间的一个接口，是认证机构CA信任范围的一种延伸。
- 注册机构RA接受用户的注册申请，获取并认证用户的身份，主要完成收集用户信息和确认用户身份的功能。
- 注册机构RA可以向其下属机构和最终用户颁发并管理用户的证书。因此，RA可以设置在直接面对客户的业务部门，如银行的营业部、机构认识部门等。
- 对于一个规模较小的PKI应用系统来说，可把注册管理的职能由认证中心来完成，而不设立独立运行的RA。但这并不是取消了PKI的注册功能，而只是将其作为认证中心的一项功能而已。
- PKI国际标准推荐由独立的RA来完成注册管理的任务，可以增强应用系统的安全。

认证中心CA的主要功能

- 接收并验证最终用户数字证书的申请；
- 证书审批，确定是否接受最终用户数字证书的申请；
- 证书签发，向申请者颁发、拒绝颁发数字证书；
- 证书更新，接收、处理最终用户的数字证书更新请求；
- 接收最终用户数字证书的查询、撤销；
- 产生和发布证书废止列表(CRL)，验证证书状态；
- 提供OCSP在线证书查询服务，验证证书状态；
- 提供目录服务，可以查询用户证书的相关信息；
- 下级认证机构证书及账户管理；
- 数字证书归档；
- 认证中心CA及其下属密钥的管理；
- 历史数据归档。

2.3 密钥管理

2.3.1 密钥对的生成

- 将私钥分发给公-私密钥对持有者系统；
- 将加密私钥分发给备份或存档系统；
- 将公钥分发给一个或多个认证机构以发放证书。

2.3.1 两种生成密钥对的系统

■ 密钥对持有者系统

- 生成密钥对的系统与储存和使用私钥的系统是同一个(可能是同一个硬件标记或软件模块)；
- 由于数字签名密钥对经常用来验证交易的不可否认性，所以采用这种方法是有充分理由的，因为私钥在其生命期内永远不会离开本地环境；
- 这种方法可以更加方便地保证任何其他的通信方都不会获得该私钥；
- 在一些规则和标准中，如ANSI X9.57标准中都规定了必须采用这种方法。

2.3.1 两种生成密钥对的系统

- 中心系统
  - 密钥对是在一些密钥管理中心系统中生成的，而私钥则被安全地分发到密钥对持有系统中;
  - 这种方法对一些象智能卡那样的密钥对持有系统来说是很必须的，因为这类系统的处理能力和储存空间都有限，要在这些设备上生成密钥对是不切实际的;
  - 中心系统可能具有更多的资源和更强的处理能力，因此可以生成高质量的密钥对，即生成的密钥对不易被预测或计算出来。
  - 密钥对中的私钥需要在中心系统进行备份或存档，因为密钥的生成和备份或存档功能是由同一系统或关系十分密切的系统来完成的。

2.3.1 两种生成密钥对的系统

- 生成密钥的两种方法都需要适应环境的变化
- 密钥对持有系统来生成数字签名密钥对；
- 中心系统采生成用于数据加密的密钥对；
- 将认证机构和私钥分开管理一定会更安全，更能抑制私钥的泄漏，而且不知晓私钥值的认证机构在为证书的主体身份、权限或其他申明提供证明时，可能更值得第三方的信任。

实例：签名证书与加密证书的制作

1. 证书的主体在个人密码设备中生成签名密钥对；
2. 从个人密码设备中导出签名公钥
3. 通过证书机构制作签名证书
4. 证书主体验证证书中的公钥的正确性

5. 证书机构在密码设备中生成加密密钥对
6. 利用证书主体的签名公钥制作数字信封导出加密私钥
7. 证书主体将加密私钥导入个人密码设备
8. **证书机构将加密私钥存档**
9. 证书机构制作加密公钥证书
10. 证书主体验证加密证书中公钥的正确性

两个私钥没有在计算机的内存中出现过

2.3.2 密钥的存储

- 公钥技术和公钥证书都要求：

  私钥的使用者只能是在相应的公钥证书中鉴定过的同一用户（个人或同一设备）。

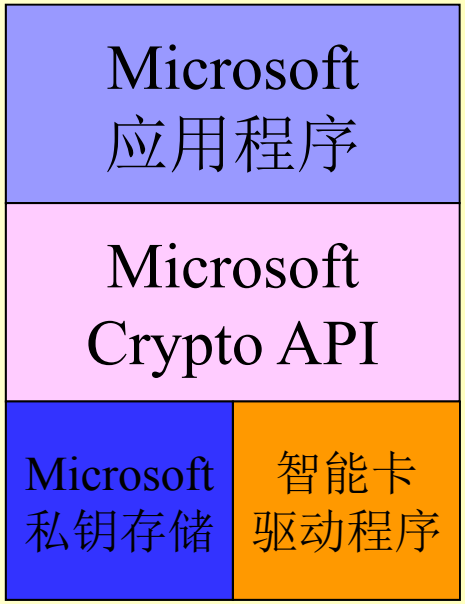
- 保护私钥的方法主要有以下几种：
  - 存储在不可写的硬件模块或标记中，如智能卡、PCMCIA卡或UsbKey；
  - 存储在计算机硬盘或其他数据存储媒介上的加密数据文件中，遵守PKCS12的PFX格式；
  - 存储在证书服务器上，当用户通过了服务器的鉴别，并在服务器上使用了一段时间后，该服务器会将私钥传送给用户。

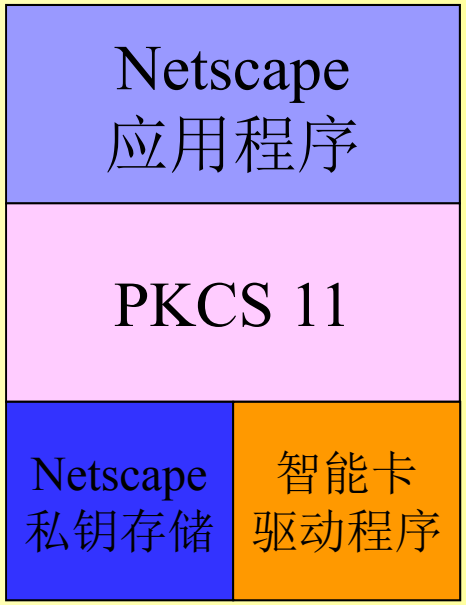
2.3.2密钥的存储

- 对私钥存储设备访问的控制
  - 用只有合法的私钥持有者才知晓的口令或PIN来计算出一个对称密钥，利用这个称密钥对私钥进行加密；
    - 用口令或个人识别号PIN来作为身份确认的主要方法；
    - 检查物理标记或生物测定法等。

2.3.2密钥的存储

- 私钥的访问





2.3.3 密钥对的更新

- 成功的安全实践表明：
  - 必须对公-私密钥对进行定期地有规律更新；
  - 另外在出现一些意外的情况，如已知或怀疑私钥被泄露时也需要更新密钥对；
  - 当一个新的密钥对生成时，需要为新的公钥生成一个新的证书；
  - 密钥对更新时，需要撤消旧的证书。

2.3.4 密钥的用途

- 数字签名密钥对管理

1. 在数字签名私钥的整个生命周期中，只有它的授权持有者才能存取它。这是为了支持不可否认性的需要。一般，可能会要求一个数字签名私钥永远不能离开生成该私钥的设备，且只能在该设备上使用和销毁，有时可能会规定必须满足这一要求。
2. 数字签名私钥一般不需要为了防止密钥意外的丢失而进行备份。如果密钥丢失了，可以很方便地生成一个新的密钥对。而且，对私钥进行备份与条件1是相冲突的。同样，数字签名私钥也不能由第三方保存。
3. 数字签名私钥不需要存档。数字签名私钥的存档与条件1也是有冲突的。事实上，当一数字签名私钥的生命周期结束时，必须将其安全销毁。因为，如果其值被泄露的话，即使很长时间没有使用它，据说它还有可能被用来伪造旧文档上的数字签名。

2.3.4 密钥的用途

- 用于支持数据加密的密钥对密钥管理
  - 数据加密私钥可能需要备份、存档成由第三方保存；
  - 备份或存档的目的是为了恢复加密信息， 因为假如密钥丢失了(例如由于设备故障或遗忘口令)，则所有使用该密钥加密的信息世部会随之丢失，而这种情况是不能接受的。
  - 当一个用于数据加密的私钥的生命周期结束时，不需要安全销毁。相反私钥是不可以销毁的。

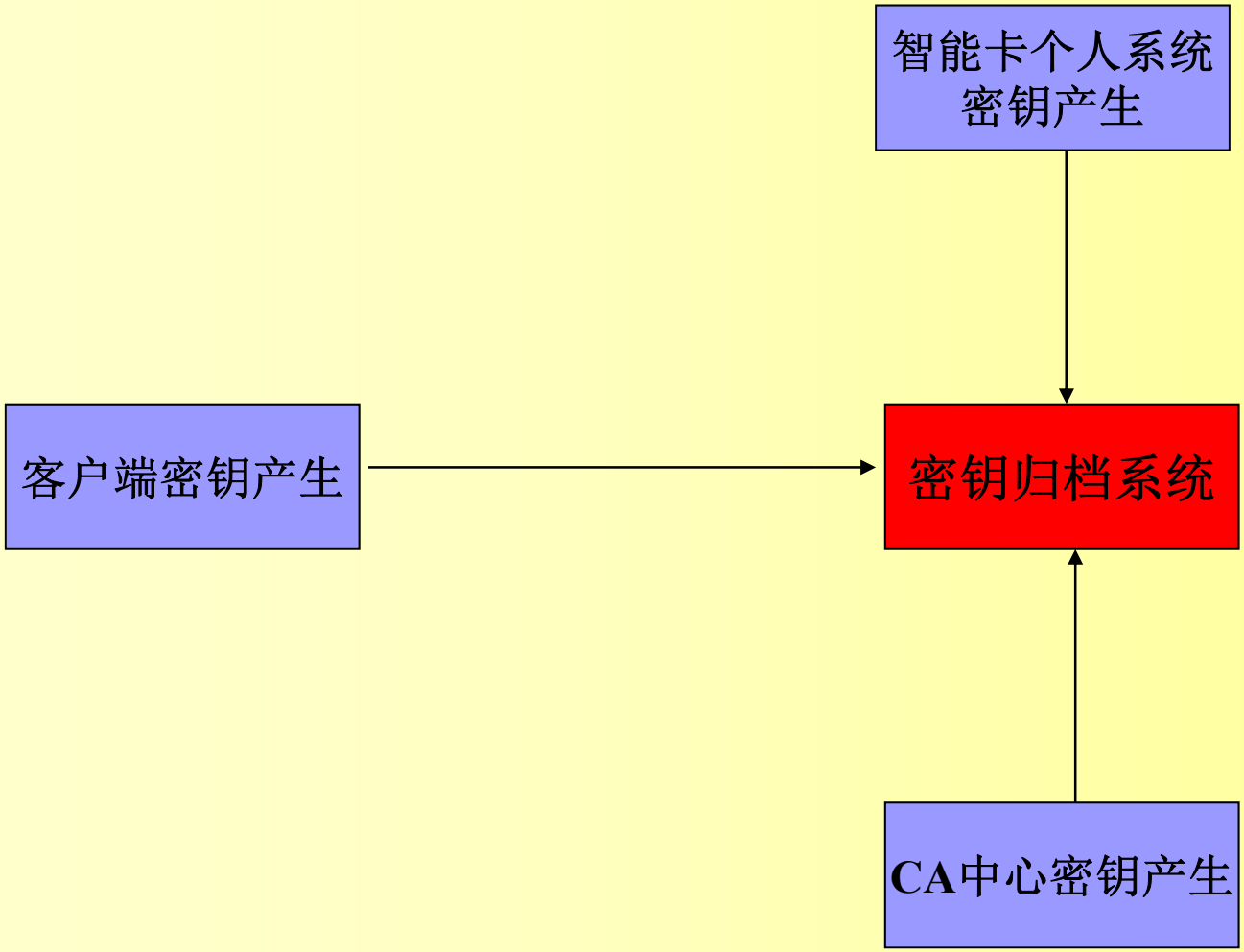
2.3.4 密钥的用途

- 用于数字签名和数据加密的公钥证书的要求是互相冲突的，所以要分离
  - 数字签名公钥证书
  - 数据加密公钥证书

2.3.5 密钥归档

- 归档的意义
  - 用户可能忘了自己的口令
  - 有的用户离开了自己的工作岗位
  - 公司需要对自己的数据具有可控性和可恢复性
  - 保密期内的加密资料，加密密钥已经不再使用
    - 2000年产生的密钥加密了2000年的资料，到2020年才到解密期，密钥到2001年就更新了。
- 归档的方法
  - 密钥的加密
  - 密钥存储的数据库的加密，例如采用(m, n)阈值方案。

2.3.5 密钥归档 — 密钥归档系统连接



2.3.5 密钥归档

- 如果实现一个中心密钥产生系统，当它为用户传递产生密钥时，它必须复制密钥并将密钥传输给密钥档案。
- 为了支持保存密钥，客户端密钥产生要求为密钥存储产生密钥的任何应用必须与密钥归档系统有接口。
  - 出现了标准协议，如CMP；
  - 安全传输，如SSL / TLS
  - PKCS#12容器。

2.3.6 密钥恢复

密钥的恢复主要是加密密钥而不是数字签名密钥的恢复。

- 当前密钥
  - 当前密钥是一个没有过期并且当前正在使用的密钥。
  - 存储私钥的设备可能物理性地损坏，但是私要没有泄漏。
- 先前的密钥
  - 每个当前密钥总有一天都会变成先前密钥。
  - 需要使用先前的密钥解密密钥更新之前的邮件。
- 旧密钥
  - 求助于离线的存储档案，访问超过一年的旧电子邮件。

2.4 证书管理

- 证书注册
- 最终实体证书更新
- CA证书更新
- 证书撤销

2.4.1 证书注册

- 初始化
  - RA名称 / 地址
  - CA证书
  - RA证书
  - 可信锚列表
  - 库名称
  - 本地密钥产生
  - 最终实体名称
- 初始信任
  - 通过证件，例如出生证、护照、驾驶证等，提供所有关于申请者身份的物理证据。
- 注册要求
  - 保护注册者的隐私：在注册过程期间利用的信息应该不在CA最终颁发的证书中发布。
- 拥有的证据
  - 确认申请者真正拥有与正在注册的公开密钥相应的私钥。

2.4.2 最终实体证书更新

- 证书更新的原因
  - 密钥有效期
  - 证书已经过期

2.4.3 CA证书更新

- 旧用旧证书
  - 原始自签名证书，此时先前的CA私钥被用来签名CA证书中先前的公开密钥。由所有依赖方拥有，这是基础。
- 新用旧证书
  - 用先前的CA私钥签名的CA证书中新的公开密钥。
  - 由先前的、已经可信的密钥证实新产生的CA公开密钥。
  - 新用旧证书的有效期限从新CA密钥的密钥产生时间开始，在所有依赖方转移到承认新密钥的适当的日期结束。最晚过期的可能时间是先前的密钥的过期期限。
- 旧用新证书
  - 用新CA私钥签名的CA证书中的原始公开密钥。
  - 在交迭期间仍然可能收到来自最终实体的证书，在最终实体上证书链被旧CA证书终止。依赖方已经转换到新密钥，旧用新允许依赖方继续信任或者确认旧CA证书。
  - 旧用新证书的有效期从先前的密钥对产生的日期开始，在旧用旧证书过期的日期结束。
- 新用新证书
  - 用新CA私钥签名的CA证书中新的公开密钥(这是新的自签名CA证书，所有依赖方使用这个CA 作为可信锚)。

2.4.4 证书撤销

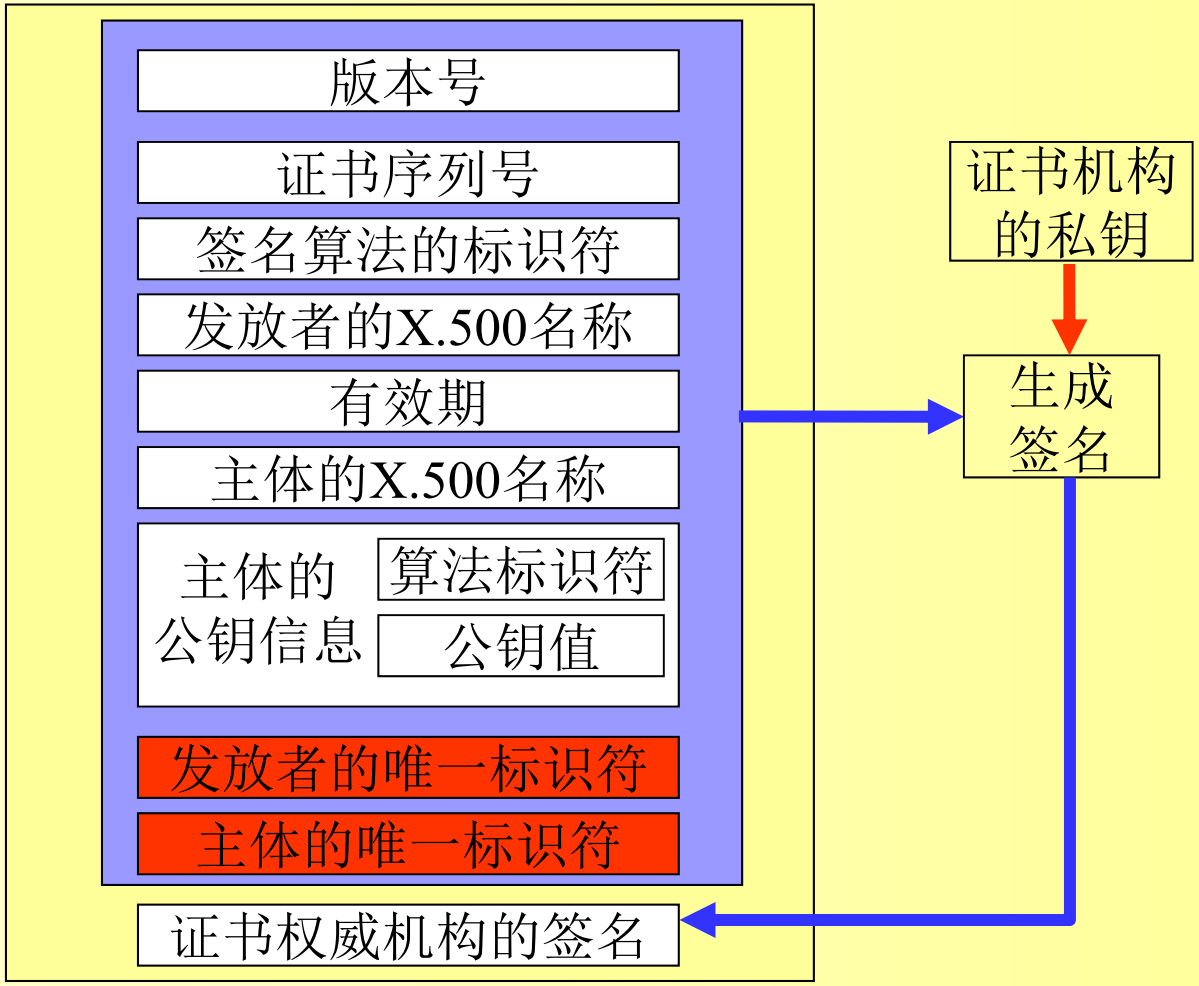
- 证书被撤销的原因
  - 当条件要求证书的有效性在证书结束日期之前终止。
  - 或者要求用户身份与私钥分离时。
- 证书撤销的要求
  - CA向向证书用户团体发布或者宣布证书已经撤销以及撤销的时间等。
- 证书撤销的有关时间
  - 撤销请求的日期。
  - 密钥泄露的实际日期。
  - 签名之类的操作仍然被认为有效的最后日期。
- 可以使用撤销机制挂起证书

3. X.509 证书格式

3.1证书 — X.509证书格式

- ITU的X.509标准定义了被人们普遍接受使用得最为广泛的标准公钥证书格式(ITU X.509标准也称为ISO／IEC 9594-8标准)。
- X.509证书格式有3个不同的标准版本：版本1格式是在1988年的第一版中定义的，版本2格式是在1993年的第二版中定义的，而版本3格式则是在1997年的第三版中定义，并在2000年的第四版中又对其进行了改进。

3.1 证书 — v1&v2证书格式



3.1 证书 — v1&v2证书格式

1. 版本：代表证书的版本格式是版本1、版本2或版本3，将来还可以是其他版本；
2. 序列号：由认证机构发放的代表该证书的唯一标识号；
3. 签名：认证机构用来对证书进行签名所使用的数字签名算法的算法标识符；
4. 证书发放者；发放证书的认证机构的X.500名称;
5. 有效期：证书的起始和终止的日期和时间;

3.1 v1&v2证书格式

6. 主体:与相应的被验证公钥所对应的私钥持有者的X.500名称;

7. 主体的公钥信息：主体的公钥值以及该公钥被使用时所用的算法标识符;

8. 证书发放者的唯一标识符：这是一个可选的位串，当不同的实体被再分配同样的名称时，利用该标识符可使发放证书的认证机构的X.500名称不具有二义性;

   （X.500名称可能被重用的情况：某个雇员名叫李进，从公司离职了，过了一段时间之后，又有一个员工也叫李进进了公司。必须避免后面的李进直接使用前面一个李进的权限。）

6. 主体的唯一标识符：这是一个可选的位串，当不同的实体被再分配同样的名称时，利用该标识符可使主体的X.500名称不具有二义性;

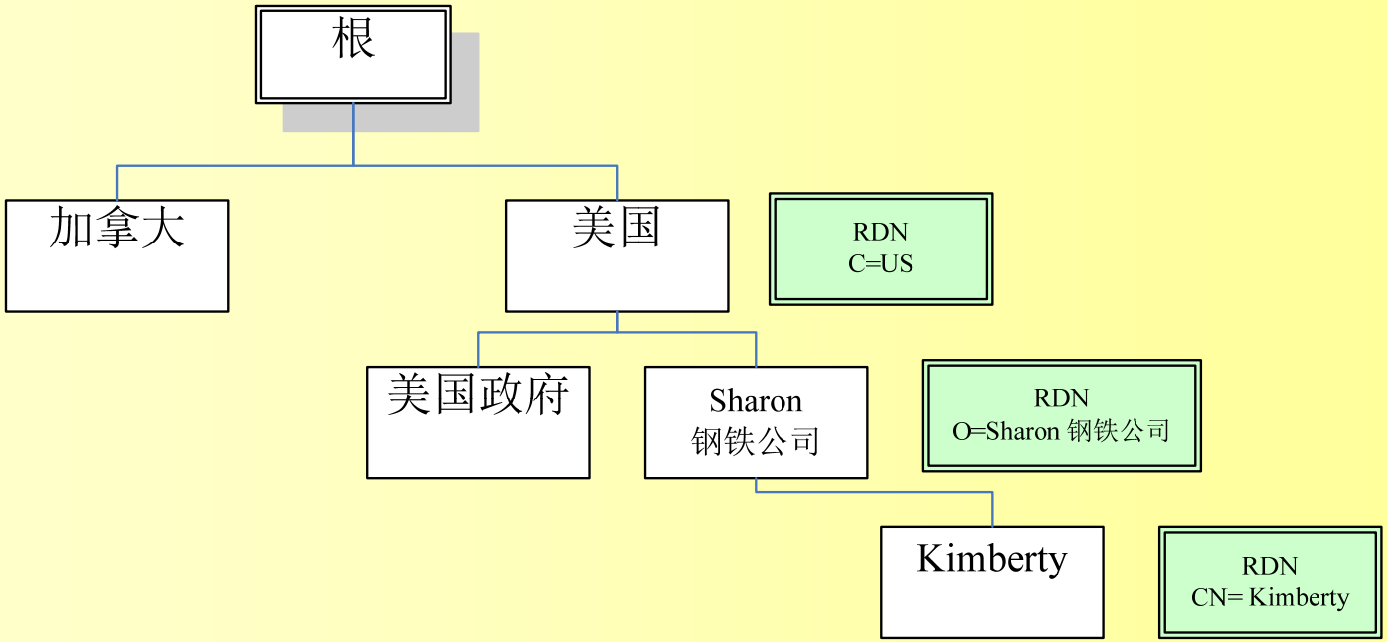
X.500名称

- 一个X.500名称由一系列目录项组成。
- 每个目录项对应现实世界中的对象，如某个人、某个组织、某个设备。
- 每个对象都有一个无二义性的名称，称为特异名（Distinguished Name, DN）。
- 对象的目录项中包含有关该对象的一系列属性值，例如某让你的电话号码、电子邮件等等。
- 为了支持无二义性命名的需要，所有的x.500目录项在逻辑上被组织成一种树形结构，称为目录信息树（Directory Information Tree, DIT）。
- 除了根节点之外，每个节点对应一个目录项，并且有一个特异名，根节点的特异名为空。

X.500名称

- 目录项的特异名由该目录项在目录信息树上的直接上级项的特异名和一个该目录项的相对特异名（Relative Distinguished Name, RDN ）联合组成。
- 实际运用时，相对特异名用一个属性值的等式说明。例如：Country = P.R.China， 组织=宝山钢铁公司，姓名=李进。

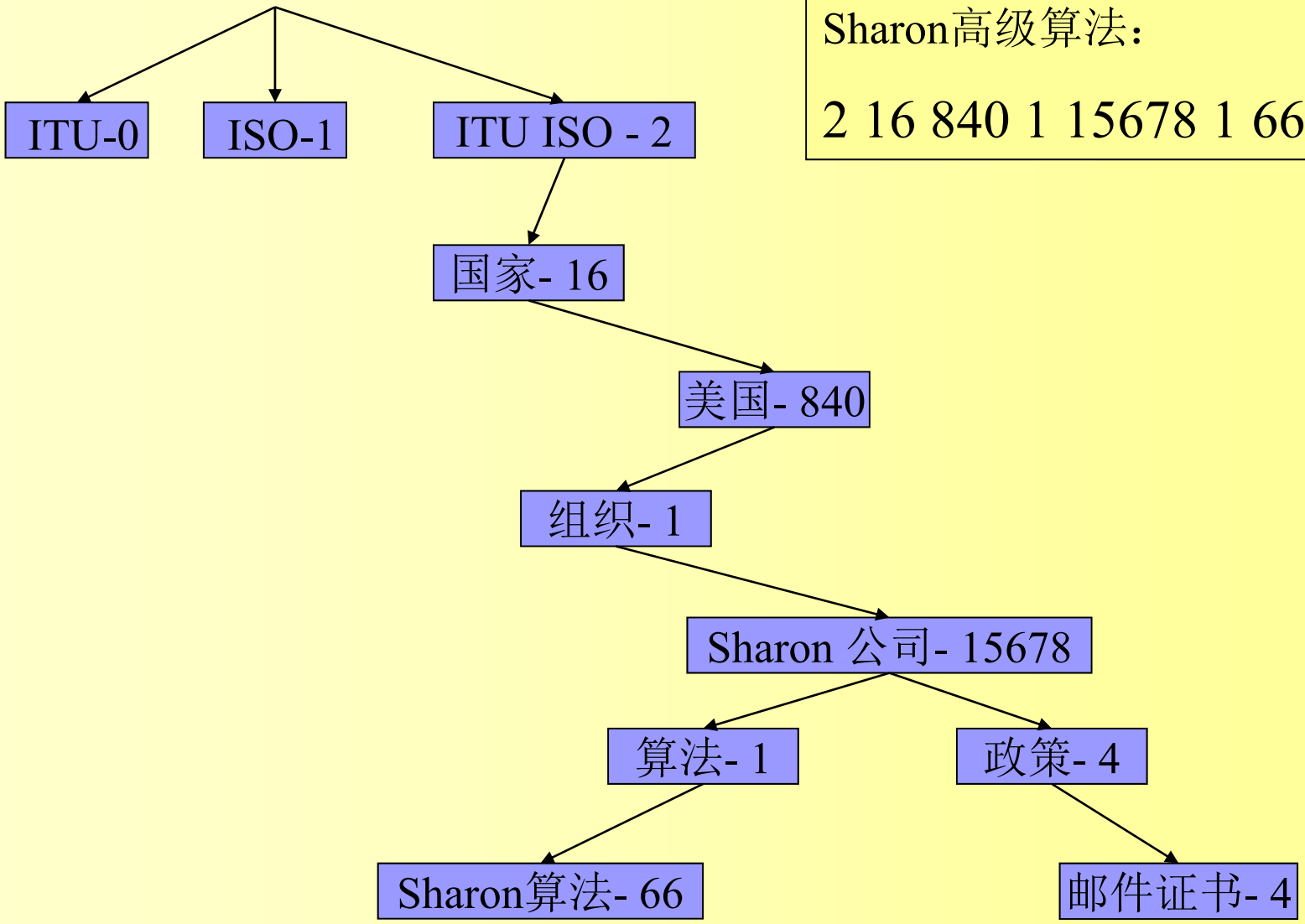
X.500的名称结构例子



对象注册

- 签名算法、签名算法、hash算法等等需要准确的标识，证书使用的政策需要标识，等等。例如
  - 用于数字签名的使用SHA-1hash 函数的DSS算法
  - 用于建立加密密钥的RSA密钥传输算法
  - 用于建立加密密钥的Diffie-Hellman密钥传输算法
- 国际对象注册机构负责公共对象的注册。
- 一个对象的标识符是由一系列整数组成，每个对象标识符都是唯一的。
- 对象标识符依据的是由一个不同的赋值机构所组成的层次结构。

对象注册



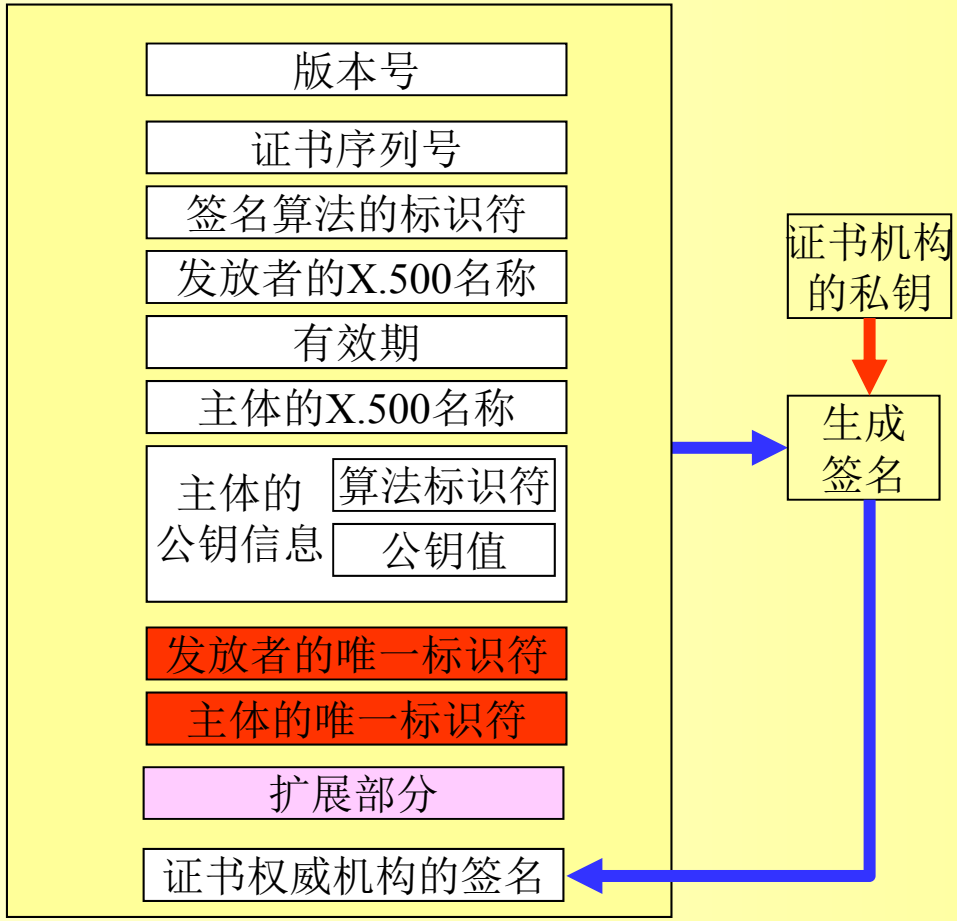
Sharon高级算法：

2 16 840 1 15678 1 66

一些公共算法标识符

- RSA安全公司的签名算法
  - **RSA的签名算法**：iso(1) 成员体(2) 美国(840) rsadsi (113549) pkcs-1 (1) sha-1 withRSAencryption （5）
  - **RSA的签名算法**： iso(1) 成员体(2) 美国(840) rsadsi (113549) pkcs-1 (1) md5 withRSAencryption （4）
  - **ANSI X9.62** ： iso(1) 成员体(2) 美国(840) X-92 (10045) 签名 (4) ecdsa-with-sha1 （1）

3.2 证书的扩展项 — X.509 证书格式v3



3.2 证书的扩展项

- 在1993至1994年间X.509证书刚开始尝试大规模商业运用时，版本1和版本2的证书格式在很多方面都显得不够完善。有很多理由表明这些证书还需要附加一些其他的信息。
  - 假设证书主体很有可能会拥有用作不同用途的不同公钥证书，并且需要定期更新密 钥，则这就需要能方便地区分同一主体的不同证书;
  - 一些应用程序需要用特殊的应用名形式而不是X.500的名称来确定用户的身份。例如在考虑电子邮件安全时，将一个公钥与电子邮件地址捆绑在一起，远比将X.500名称与电子邮件地址捆绑在一起来得重要;
  - 不同的证书可能是在不同的政策和实践准则下发放的，这些政策和实践准则决定了公钥用户对证书的信任程度。

3.2 证书的扩展项

- 例如，在发放一个用于对普通电子邮件进行加密的证书时，认证机构可能不进行所有的身份检查和授权检查；
- 但如果该证书是用来验证企业与企业间巨额金融交易的数字签名的，则进行身份检查和授权检查就是必须的了；
- 在证书格式中需要添加一些附加字段；
- 为了满足已知的和将来未知的需求，显然需要不断地在证书中增加各种不同的字段；
- 负责该领域工作的标准化组织(ISO / IEC、ITU和ANSIX9)在证书中用一个共同的位置，来增添了一个通用扩充机制，其结果就是X.509版本3的证书格式。

3.2扩展字段的基本概念—扩展项格式

- 扩展类型
  - 每个扩展字段都必须有一个需要被注册的类型，如同注册算法对象一样，将一个标识符赋予该类型。
  - 公共扩展类型必须为所有的实施者所了解。
  - 团体可以定义自己的扩展类型。
- 关键值
- 扩展字段值

```
Extension ::=  SEQUENCE {
    extnId   EXTENSION.&id ({ExtensionSet}),
    critical   BOOLEAN DEFAULT FALSE,
    extnValue  OCTET STRING
}
```

3.2 扩展字段的基本概念—非关键

- 如果关键程度指示器指明一个扩展类型是非关键的，那么当使用证书的系统无法识别该扩展类型时，允许忽略该扩展字段。
  - 假设某个扩展字段计划用来给一组特殊的应用程序传送证书主体的可选形式名称。这样的字段可以被标记为非关键的，因为其他不使用可选形式名称的应用程序仍可以基于原来的主体名称字段有效地使用证书，即使这些应用程序并不理解可选名称扩展。

3.2 扩展字段的基本概念—关键

- 而如果关键程度指示器指明一个扩展类型是关键的，则任何对该证书的部分使用都是不安全的，除非系统能识别扩展类型并调用与之关联的函数。
  - 假设某一扩展字段传送的信息是为了将证书用途限制在认证机构为其规定的范围内。
  - 如果某一使用证书的系统不理解该扩展字段并忽略了该字段，就有可能执行不安全的操作。
  - 因此，认证机构必须将该扩展字段的关键指示器标记为关键。

3.2 证书的扩展项 — X.509 v3的命名

- 版本3不再局限于用X.500名称来确定如证书主体和证书发放者那样的实体的身份;
- 任何一个实体都可以用一个或多个不同形式的名称来确定;
- Sharon钢铁公司的首席执行官Kimberly可能具有下列不同形式的名称：
  - X.500名称: {国家=US，组织=Sharon钢铁公司，名字=Kimberly);
  - X.500名称: {国家=US，组织=Sharon钢铁公司，职位=CEO};
  - 电子邮件地址：Kimberly@sharons.com

3.2 证书的扩展项 — X.509的名称形式

- 因特网电子邮件地址。
- 因特网域名;
- X.400电子邮件地址;
- X.500目录名;
- EDI通信方的名称(由一个名称赋值机构加上该机构所赋的通信方名称组成);
- Web统一资源标识符(URL)。
- 因特网IP地址。

3.2 证书的扩展项 — 标准证书扩展

- ISO / IEC、ITU和ANSI X9等标准组织制定了一系列X.509版本3的证书扩展标准，标准的扩展可以分成如下几个组：
  - 密钥信息；
  - 政策信息；
  - 主体与发放者的属性；
  - 认证路径约束；
  - 与证书撤消表(CRLs)相关的扩展。

3.2 密钥信息扩展 — 机构密钥标识符

CA的密钥对也需要定期更换，因此CA签发的证书也需要有一个域来指示验证该证书的CA证书。

该扩展项用于区分由发放证书的认证机构所使用的不同的证书签名密钥(例如，在不同时间间隔使用的不同密钥、在同一时间使用不同密钥)。

- 使用显式密钥标识符;
- 使用CA证书的颁发者名称和证书编号;
- 或者同时使用密钥标识和证书编号；

```
authorityKeyIdentifier EXTENSION ::= {
        SYNTAX AuthorityKeyIdentifier
        IDENTIFIED BY id-ce-authorityKeyIdentifier
}
AuthorityKeyIdentifier ::= SEQUENCE {
        keyIdentifier [0] KeyIdentifier
        authorityCertIssuer [1] GeneralNames
        authorityCertSerialNumber [2] CertificateSerialNumber
```

3.2 密钥信息扩展 —主体密钥标识符

- 该扩展项用于区分由同一证书主体使用的不同密钥。例如，主体可能会定期地更换其密钥对，该扩展项用以指出按顺序哪个公钥应在某一给定的证书中被认证。

3.2 密钥信息扩展 — 密钥用途

- 该扩展项用以说明密钥的用途，如数字签名、邮件签名证书、小额电子银行证书、大额电子银行证书、不可否认、密钥加密、数据加密、Diffie-Hellman 密钥协议、证书签名、CRL签名、加密或解密。
- 证书的用户必须要在证书的某个域中获得（CA也需要在这个域中规定）该证书的用途（与证书的策略相关）。
- 通过对象标识符来标识密钥用途，一个密钥用途的标识符由ITU X.600 或 ISO/IEC 9834-1定义。

```
extKeyUsage EXTENSION ::= {
        SYSTAX          SEQUENCE SIZE （1…MAX) OF KeyPurposeId
        IDENTIFIED BY   id-ce-extKeyUsage }
KeyPurposeId ::= OBJECT IDENTIFIER
```

3.2 密钥信息扩展 — 密钥用途

```
keyUsage EXTENSION ::= {
    SYNTAX KeyUsage
    IDENTIFIED BY id-ce-keyUsage
 }

KeyUsage ::= BIT STRING {
    digitalSignature (0),
    nonRepudiation (1),
    keyEncipherment (2),
    dataEncipherment (3),
    keyAgreement (4),
    keyCertSign (5),
    cRLSign (6),
    encipherOnly (7),
    decipherOnly (8)
}
```

3.2 证书的扩展项 —政策信息扩展

- 证书政策
  - 该扩展项用于说明认证机构对证书所规定的政策和操作说明，同时还表示了可选的政策限定符；
- 政策映射
  - 这个扩展项只能用于CA 证书，借助该扩展项，证书发放者可以指明发放者的证书政策等价于主体认证机构领域使用的一个政策。

3.2 证书的扩展项 —主体及发放者属性的扩展

- 主体备用名
  - 该扩展项包含证书主体的一个或多个备用的、无二义性的名称，这些名称可以使用任何一种形式。它们可能不使用，X.500名称而使用自己的名称形式。
- 发放者备用名
  - 该扩展项包含证书发放者的一个或多个备用名。名称形式与主体备用名扩展相同。该扩展项对那些必须通过某些在特殊应用环境如Web或e-mail中非常有意义的名称来确定或识别身份的认证机构特别有用；

3.2 证书的扩展项 —认证路径的约束扩展

- 基本约束
  - 该扩展项用以指明证书主体是否可以充当一个认证机构，或仅仅是作为一个最终实体而已。此标识非常重要，它可以防止最终用户错误地或欺诈性地冒充认证机构；
  - 若某个主体可以充当认证机构，则该标识还规定了认证路径的长度约束；例如：某个机构只可以给最终用户发放证书，而不能给机构发放正式。

3.2 证书的扩展项 —认证路径的约束扩展

- 命名约束
  - 该扩展项用于限制从本证书出发的认证路径上的后续证书可以接受的名称空间。
  - 该机制允许任何认证机构在验证另外一个认证机构的时候，去确切指定在认证路径中什么样的名称可以用在后继的证书中。
  - 例如，当Dannille机器制造公司的认证机构交叉认证Sharon钢铁公司的中央认证机构时，Dannille的认证机构可以指明出现在后继认证路径中的可接受主体名必须是下列子树中的X.500名称：
    - {国家=美国，组织= Sharon钢铁有限公司……}
    - {国家=英国，组织= Sharon钢铁有限公司……}
  - 这并不是说Sharon可以签发的证书仅有这么几个名称， Sharon的认证机构完全可以给其他的外国子公司签发证书，甚至可以给与之有生意来往的其他的不同的公司签发证书。但Dannille公司希望将它的认证限定在两个标识出的子树上，因为这两个子树仅包括了Sharon公司中与Dannille公司有生意往来的人。 Dannille公司通过不允许认证路径通往那些它所不知道也不需要知道的Sharon公司的其他部分来减少自己的风险。

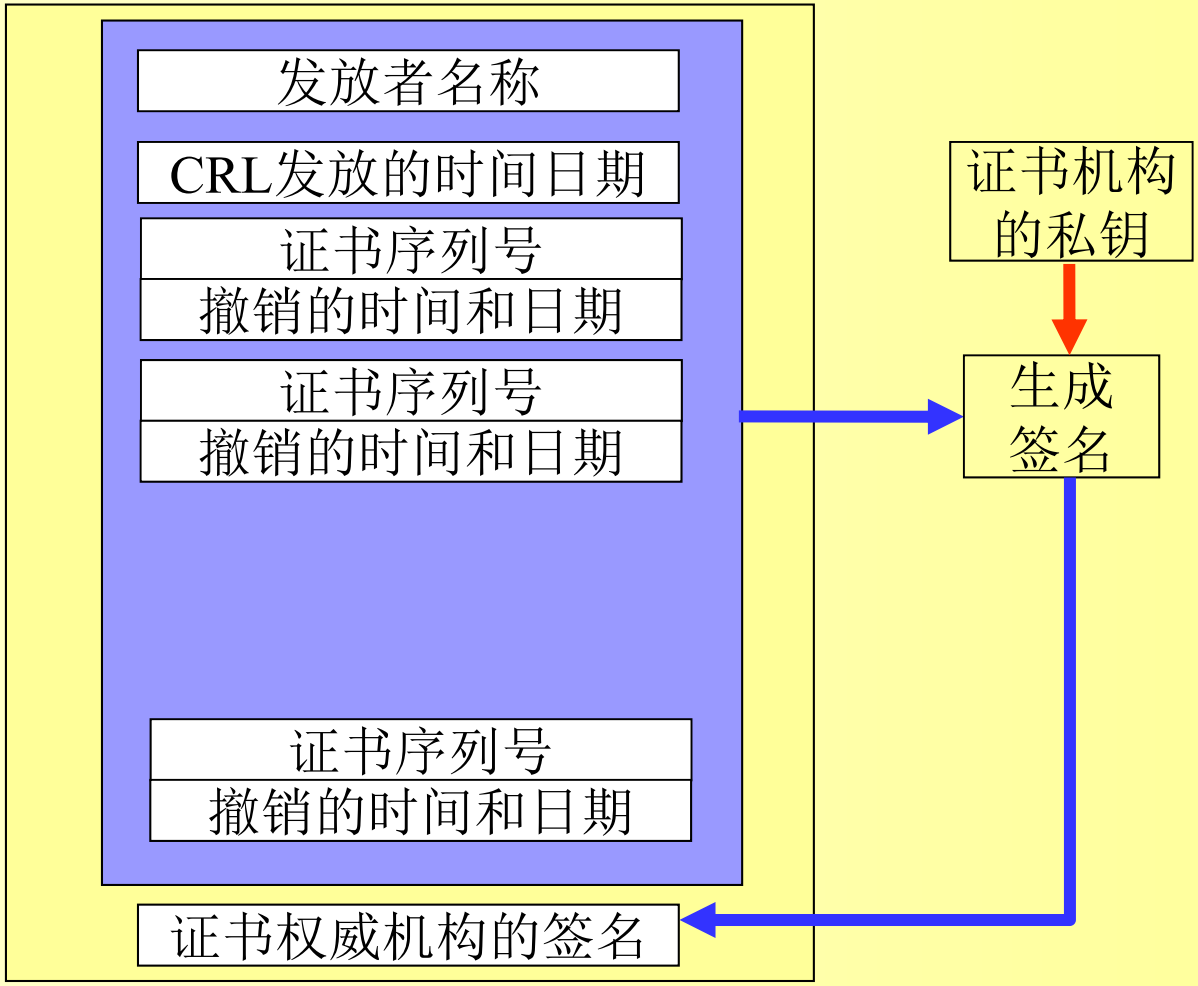
3.3 证书撤销

- 公钥证书具有有限的有效生命期；
- 某些情况发生时时证书用户必须在证书期满前停止对证书的信任：
  - 私钥泄露；
  - 名称变更；
  - 与认证机构的关系发生变化；
  - ……
- 这时，认证机构可以撤消证书。由于证书可能被撤消，所以证书的运作期可能比原定的有效期要短。

请求证书撤消

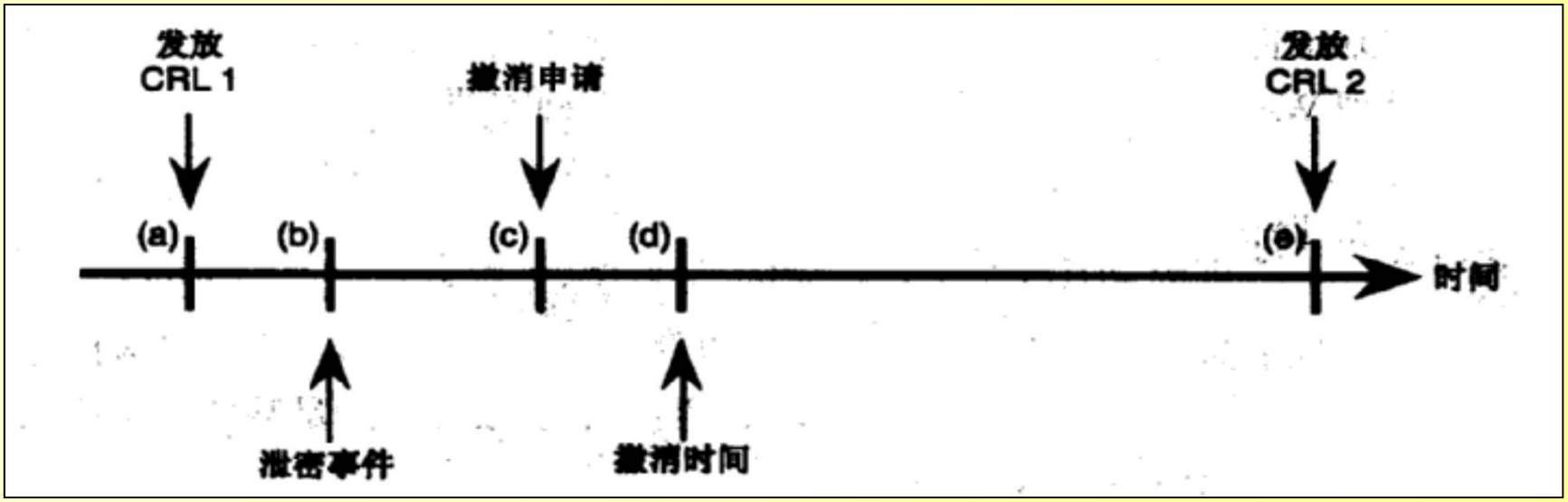
- 谁有权撤消证书依赖于认证机构的准则，包括认证机构用户在内的通信各方都要了解这些准则。
- 认证机构用户有权请求撤消自己的证书。
- 在某些规定的情况下，例如用户违反了职责或用户死亡时，认证机构官员有权撤消用户的证书。
- 其他人也可能有权请求撤消证书，例如，用户的雇主可以请求撤消雇佣关系证书。

3.2.4 撤消表的格式


发放者名称
CRL发放的时间日期
证书序列号
撤销的时间和日期
证书序列号
撤销的时间和日期
证书序列号
撤销的时间和日期
证书权威机构的签名
证书机构的私钥
生成签名

3.2.4 证书撤消过程的时间流

- (a) 发放第一份证书撤消表;
- (b) 密钥泄漏；
- (c)证书撤消请求；
- (d) 撤销时间；
- (e) 发放第二份证书撤销表。



3.2.4 证书撤消过程的时间流

- (b)—(c)时段：这时私钥已被泄露；但还未通知认证机构，这时无法期望信任方知晓私钥已被泄露，对用户来说可能知道也可能不知道私钥的泄密，这一时期由用户来承担私钥滥用的最大风险可能是比较合理的；
- (c)—(d)时段：这时认证机构已收到私钥被泄露的通知，但是还未分发证书撤消表，这时无法期望证书用户一定知晓私钥的泄露，这一时期由认证机构来承捐私钥滥用的最大风险可能是比较合理的；

3.2.4 证书撤消过程的时间流

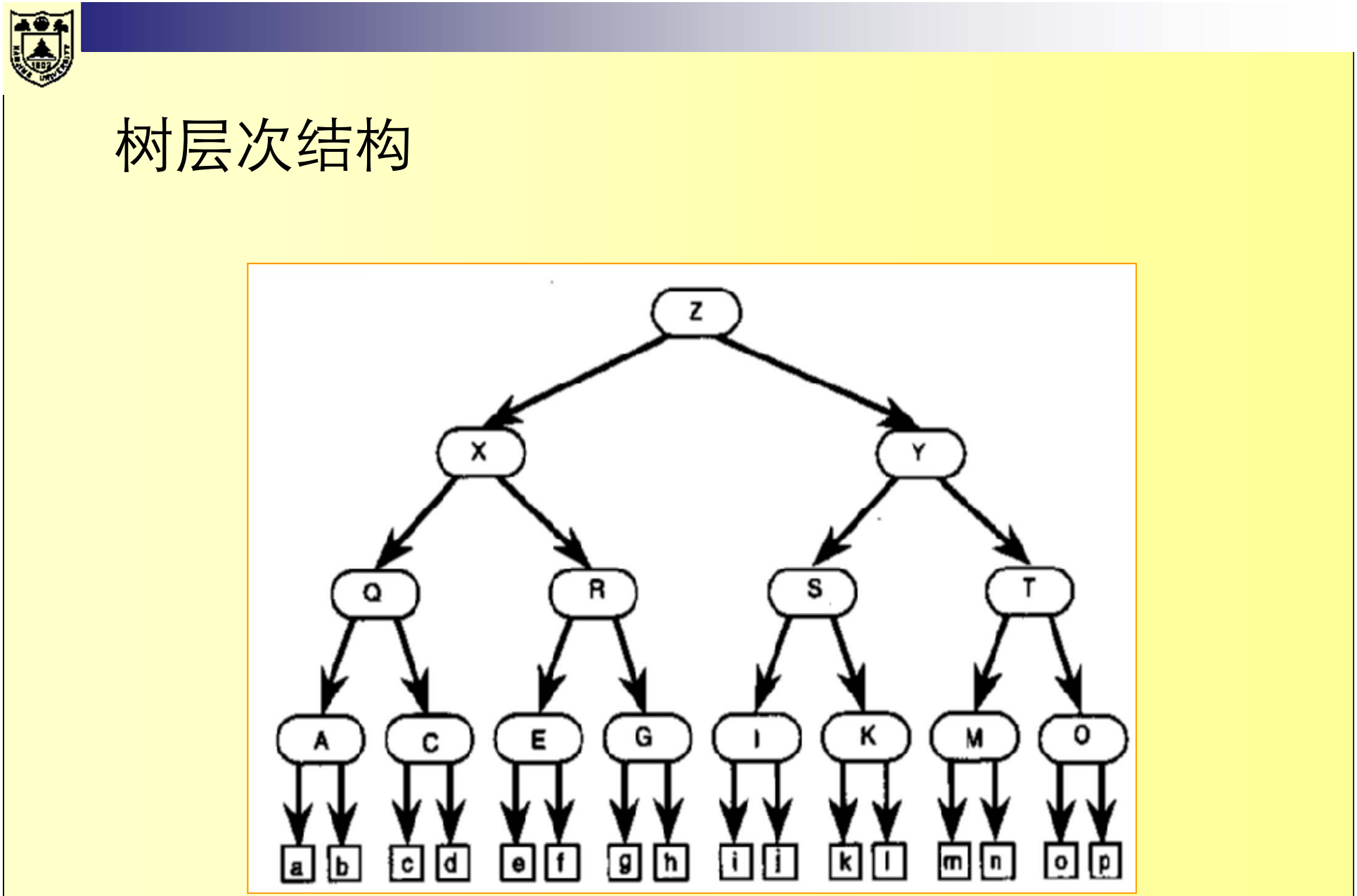
- (d)—(e)时段：这时认证机构发出了证书撤消通知，但信任方还不知道证书已撤消；
  - 这一时期风险的分摊依赖于所使用的特定证书撤消机制对采用周期性证书撤消表的证书用户来说，第二份证书撤消表CRL2发放前不会知道证书已被撤消；
- (e)时刻后：这时认证机构已完成了证书撤消信息的发布，若某一证书用户这时仍在使用已撤消的证书，则由该证书用户来承担相应的后果尸般是比较合理的。

4. 信任模型

- 为了使用某个异地通信方的公钥，证书用户(使用方)必须找到一条有效的完整的认证路径，将公钥从一个或多个认证机构传送到可信任的根认证机构——证书用户持有该CA的公钥并信任该CA。
- 在建立庞大的可缩放的PKI过程中，一个主要的挑战就是如何使寻找有效认证路径的过程变得简单、方便和高效。
- 这在很大程度上要依救于构造CA间结构关系的规则或协定，因为借助CA间的结构关系，才能使得一些认证机构能够验证其他认证机构的身份。

树层次结构

- 把一个由最终实体组成的庞大团体中的各个成员通过可接受的简短路径来跟一小部分可信任的根认证机构联系起来；
- 通过其中的每条路径可以通往一系列可信任的认证机构；
- 数学图论中的树型结构或称为层次型结构恰好为我们提供了解决这一问题的最佳方法。



树层次结构

树层次结构

- 所有的认证路径都是从根认证机构z开始；
- 证书用户必须把根认证机构作为其唯一最终可信任者，换言之，他们必须持有根认证机构公钥的可靠副本，并且通过独立的途径使其生效。
- 寻找一条通向任一个最终实体的认证路径是很容易的，比如任何证书用户都可以通过使用一个由4份证书所组成的认证路径来取得a公钥的有效副本：
  - 由z为X签发的证书(证书用户固有地相信z的公钥)；
  - 由x为Q签发的证书；
  - 由Q为A签发的证书；
  - 由A为最终实体a签发的证书。

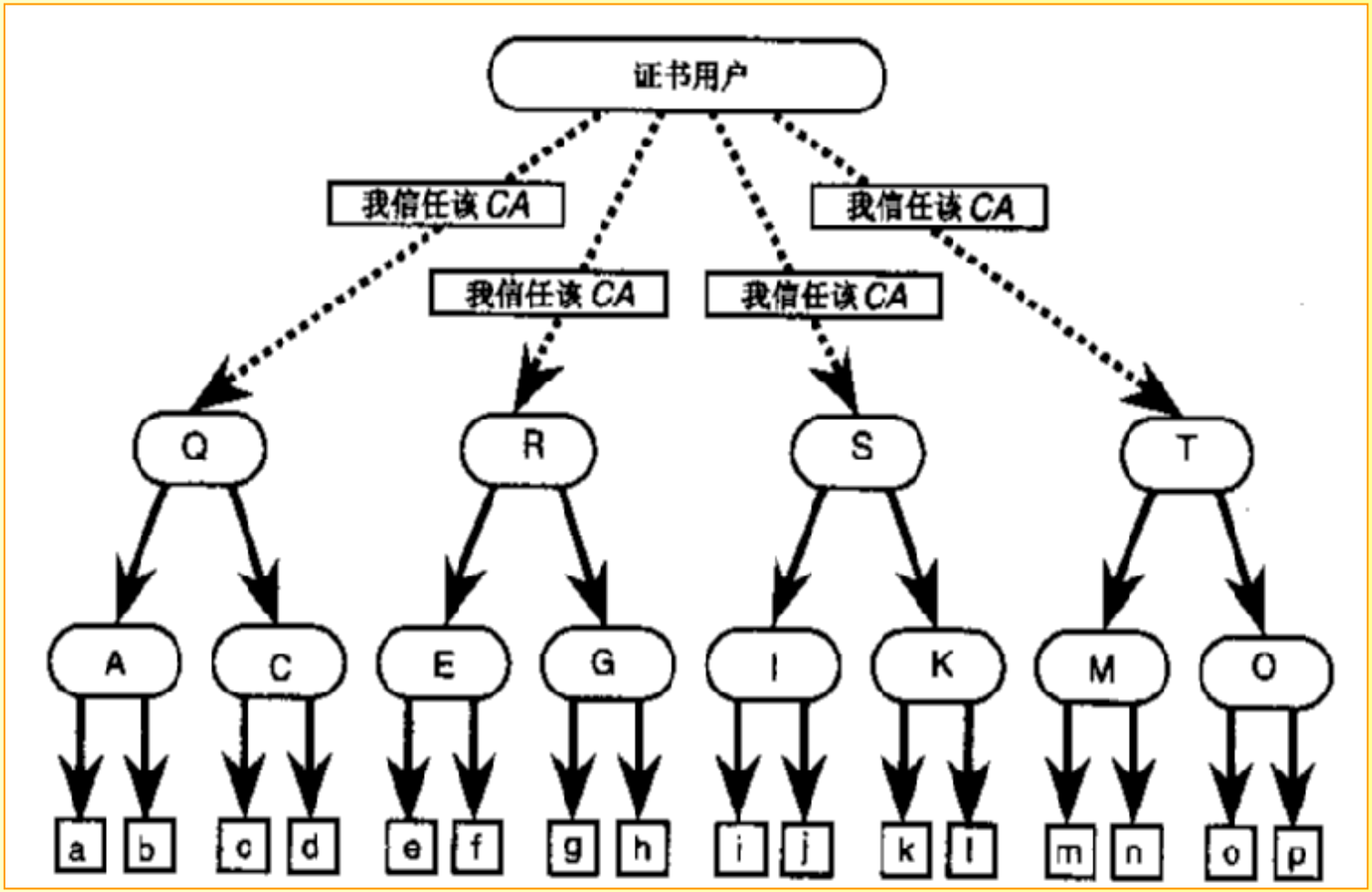
树层次结构

- 在使用任何认证路径时，证书用户必须相信该认证路径上的每一个认证机构都已经履行了忠实义务，并且已经采取了适当的防范措施，以保证排除那些自称是来自认证机构的伪造的认证证书。
- 在这个树型模型里，所有的参与方都必须承诺恪守PKI杜团所公认的行为准则。

森林型结构

- 在树型结构的PKI中，在为某个特定的团体，如在为企业的雇员团体、为B2B交易的客户团体以及为某个垂直型行业的联合团体设计PKI时并不难做到；
- 但在现实的安全应用环境中，要为所有的参与方建立一个包容万象的树型结构并不总是可行的。

森林型结构


证书用户
我信任该 CA
我信任该 CA
我信任该 CA
我信任该 CA
Q
R
S
T
A
C
E
G
I
K
M
O
a
b
c
d
e
f
g
h
i
j
k
l
m
n
o
p

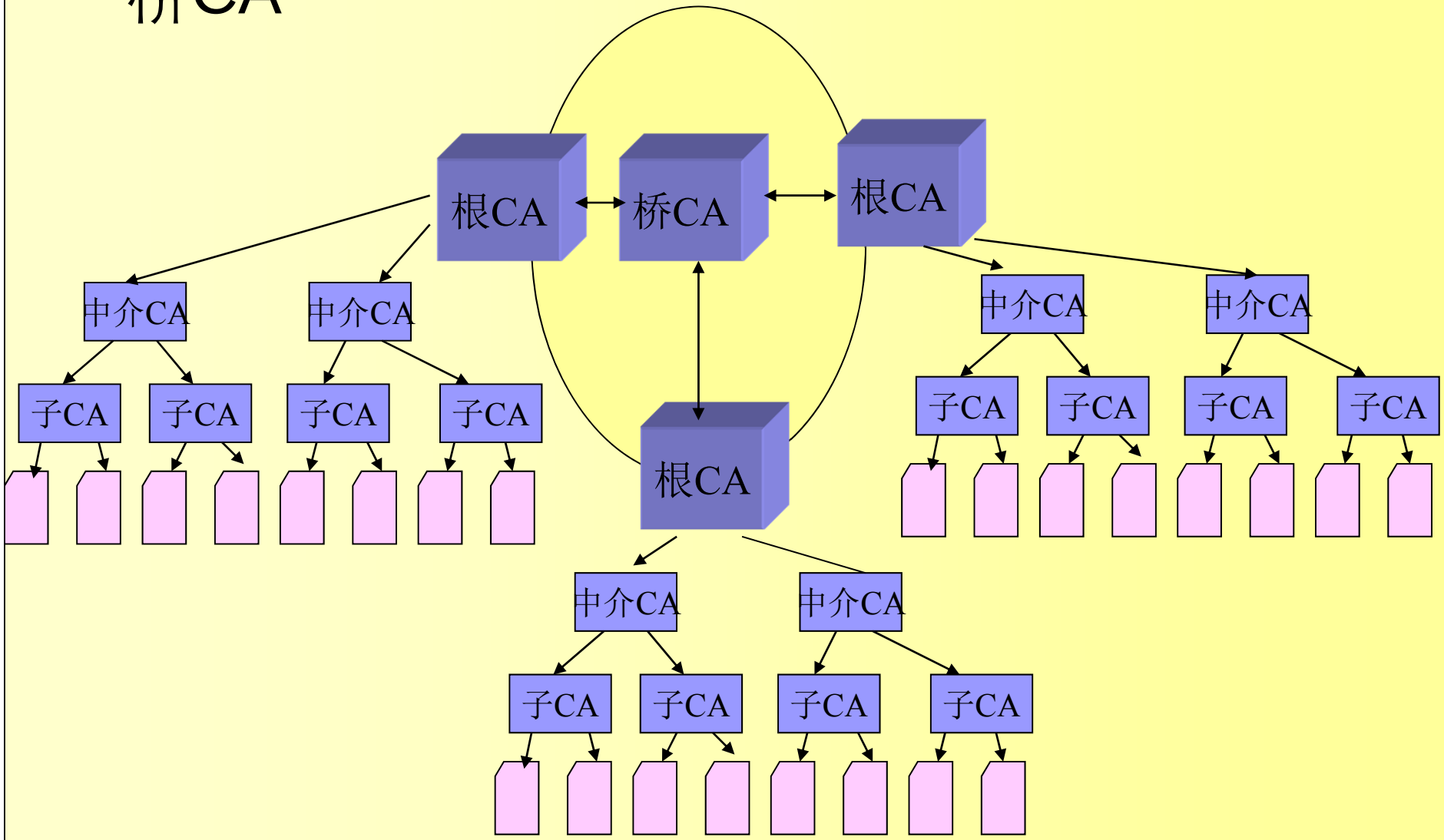
森林型结构

- 在建立树型结构中可信任的根节点时可以有各种机制，这些机制主要有：
  - 直接由用户控制 用户在使用证书应用系统时，可以自行决定可信任的根认证机构；
  - 直接由领域控制 某些领域的管理员会强令该领域内的所有证书使用系统(如浏览器或邮件客户机)接受一批特定的可信任根认证机构。

交叉认证

- 交叉认证是一种把以前无关的CA连接在一起的有用机制，从而使得在它们各自主体群之间的安全通信成为可能。
- 交叉认证中的主体和颁发者都是CA
- 当这个区别是重要的时候，
  - 如果两个CA属于相同的域，这种处理被称作域内交叉认证。
  - 如果两个CA属于不同的域(例如，当在一家公司中的CA认证了在另一家公司中的CA)，这个处理被称作域间交叉认证。
  - 交叉认证可以是单向的，也可以是双向的。

桥CA


根CA
桥CA
根CA
中介CA
中介CA
中介CA
中介CA
子CA
子CA
子CA
子CA
子CA
子CA
子CA
子CA
根CA
中介CA
中介CA
子CA
子CA
子CA
子CA

实践课题建议

1. 利用CryptAPI接口编写公钥证书的应用的程序
   - 数字信封的加密
   - 数字签名和签名验证
2. 在Windows 2003、Windows2008等服务器版本的操作系统上
   - 安装证书服务器
   - 制作证书
   - 编写基于证书的加解密、数字签名。
3. 编写一个制作证书的程序