上讲主要内容

- 数字签名的简介
- 基于**RSA**数字签名
- 基于离散对数数字签名
  - **ElGamal**数字签名
  - **Schnorr**数字签名
  - DSA数字签名
- 基于**ECC**数字签名

密钥管理技术

主讲人：马春光

Email：machunguang@hrbeu.edu.cn

主要内容

- 密钥管理的简介
- 密钥的生命周期
- 公钥证书
- 密钥分配
- 密钥协商
- 密钥托管
- 密钥分割

引言

密钥是密码系统中的可变部分。现代密码体制要求密码算法是可以公开评估的，整个密码系统的安全性并不取决对密码算法的保密或者是对密码设备等保护，决定整个密码体制安全性的因素是密钥的保密性。也就是说，在考虑密码系统的设计时，需要解决的核心问题是密钥管理问题，而不是密码算法问题，密钥管理是密码学许多技术（如机密性、实体身份验证、数据源认证、数据完整性和数据签名等）的基础，在整个密码系统中是极其重要的，密钥的管理水平直接决定了密码的应用水平。

历史表明：从密钥管理途径窃取秘密要比单纯从破译密码算法窃取秘密所花费的代价要小得多。

密钥管理的简介

密钥管理就是在授权各方之间实现密钥关系的建立和维护的一整套技术和程序。密钥管理是密码学的一个重要分支，也是密码学最重要、最困难的部分，在一定的安全策略指导下完成密钥从产生到最终销毁的整个过程，包括密钥的生成、存储、分配和协商、使用、备份/恢复、更新、撤销和销毁等。

密钥管理是一门综合性的系统工程，要求管理与技术并重，除了技术性的因素外，还与人的因素密切相关，包括密钥管理相关的行政管理制度和密钥管理人员的素质。密码系统的安全强度总是取决于系统最薄弱的环节（木桶原理），因此，再好的技术，如果失去了必要管理的支持，终将使技术毫无意义。管理只能通过健全相应的制度以及加强对人员的教育、培训来解决。

密钥管理的目的

- 保证密码系统对密钥的使用需求，及时维护和保障密钥；
- 对密钥实施有效的管理，保证密钥的*绝对*安全。

密钥管理的原则

策略是密钥管理系统的高级指导，而机制是实现和执行策略的技术机构和方法。

- 明确密钥管理的策略和机制
- 全面安全原则

指必须在密钥的产生、存储、分发、装入、使用、备份、更换和销毁等全过程中对密钥采取妥善的安全管理。

- 最小权利原则

指只分配给用户进行某一事务处理所需的最小的密钥集合。

- 责任分离原则

指一个密钥应当专职一种功能，不要让一个密钥兼任几种功能 。

密钥管理的原则(续)

- 密钥分级原则

指对于一个大的系统，所需要的密钥的种类和数量都很多，根据密钥的职责和重要性，把密钥划分为几个级别。

- 密钥更换原则

指密钥必须按时更换。否则，即使采用很强的密码算法，只要攻击者截获足够多的密文，密钥被破译的可能性就非常大。

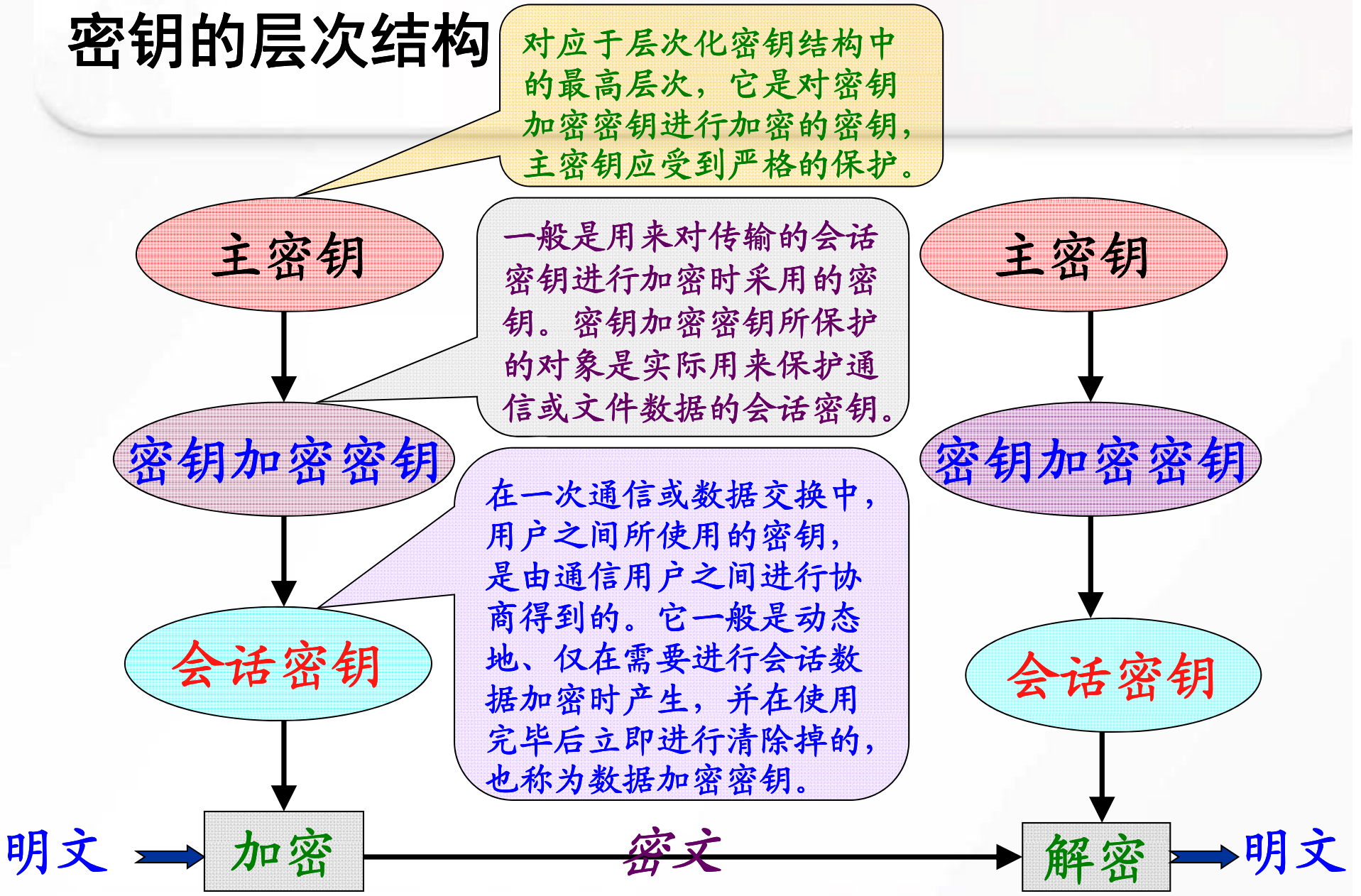
- 密钥应有足够的长度

密码安全的一个必要条件是密钥有足够的长度。

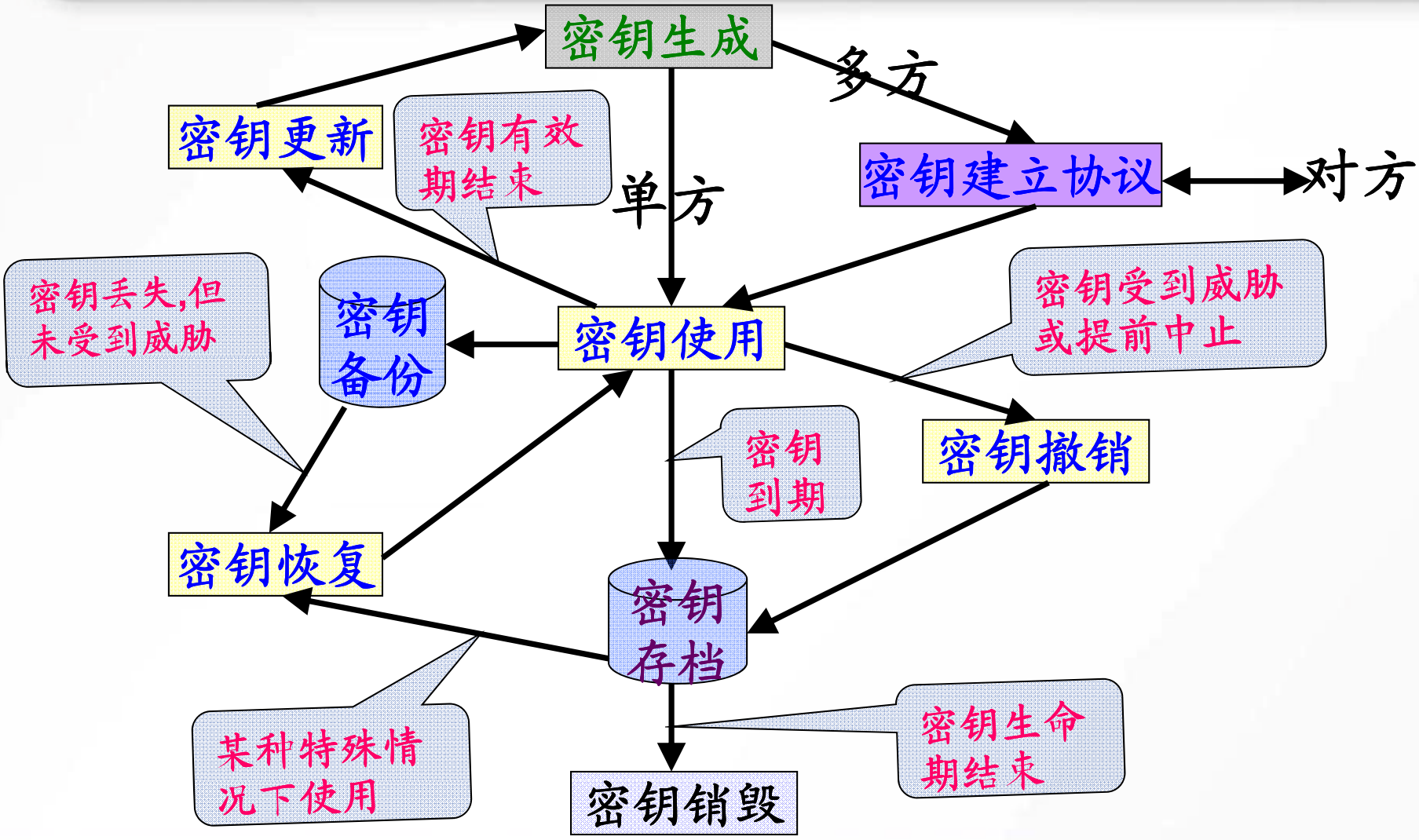
- 密钥体制不同，密钥管理也不相同

由于传统密码体制与公开密钥密码体制是性质不同的两种密码，因此它们在密钥管理方面有很大的不同。

密钥的层次结构


对应于层次化密钥结构中的最高层次，它是对密钥加密密钥进行加密的密钥，主密钥应受到严格的保护。
主密钥
一般是用来对传输的会话密钥进行加密时采用的密钥。密钥加密密钥所保护的对象是实际用来保护通信或文件数据的会话密钥。
密钥加密密钥
在一次通信或数据交换中，用户之间所使用的密钥，是由通信用户之间进行协商得到的。它一般是动态地、仅在需要进行会话数据加密时产生，并在使用完毕后立即进行清除掉的，也称为数据加密密钥。
会话密钥
主密钥
密钥加密密钥
会话密钥
明文
加密
密文
解密
明文

密钥的生命周期


密钥生成
多方
密钥更新
密钥有效期结束
单方
密钥建立协议
对方
密钥丢失,但未受到威胁
密钥备份
密钥使用
密钥受到威胁或提前中止
密钥到期
密钥撤销
密钥恢复
密钥存档
某种特殊情况下使用
密钥生命期结束
密钥销毁

密钥状态

- **使用前状态：** 密钥不能用于正常的密码操作。
- **使用状态：** 密钥是可用的，并处于正常使用中。
- **使用后状态：** 密钥不再正常使用，但为了某种目的对其进行离线访问是可行。
- **过期状态：** 密钥不再使用，所有的密钥记录已被删除。

密钥的生命周期

- 生成
- 存储
- *建立*
- 使用
- 备份/恢复
- 更新
- 存档/撤销/销毁

密钥的生成

- 密钥的大小与产生机制直接影响密码系统的安全，所以，对于一个密码体制，如何产生好的密钥是很关键的，密钥生成是密钥生命周期的基础阶段。
- 密钥的生成一般首先通过密钥生成器借助于某种噪声源产生具有较好统计分析特性的序列，以保障生成密钥的随机性和不可预测性，然后再对这些序列进行各种随机性检验以确保其具有较好的密码特性。
- 用户可以自己生成所需的密钥，也可以从可信中心或密钥管理中心申请，密钥长度要适中，但要能够抵御穷举攻击。
- 不同的密码体制或密钥类型，其密钥的具体生成方法一般是不相同的，与相应的密码体制或标准相联系。

密钥的存储

- 密钥的安全存储实际上是针对静态密钥的保护；

  如果密钥不是在使用时临时实时产生并一次使用，则必然要经历存储的过程。

  其目的是确保密钥的秘密性、真实性以及完整性。

- 对静态密钥的保护常有两种方法：

  - 基于口令的软保护；

    文件形式或利用确定算法来保护密钥。

  - 基于硬件的物理保护；

    存入专门密码装置中(如**ICCard**、**USB Key**、加密卡等)。

    安全可靠的存储介质是密钥安全存储的物质条件，安全严密的访问控制是密钥安全存储的管理条件。

密钥的使用

利用密钥进行正常的密码操作，如加密、解密、签名、验证等，通常情况下，密钥在有效期之内都可以使用。应注意使用环境对密钥的安全性的影响。

密钥安全使用的原则是不允许密钥以明文的形式出现在密钥设备之外。

密钥的备份/恢复

- **密钥备份：** 指密钥处于使用状态时的短期存储，为密钥的恢复提供密钥源。要求安全方式存储密钥，并且具有不低 于正在使用的密钥的安全控制水平。
- **密钥恢复：** 当密钥因为人为的操作错误或设备发生故障时可能发生丢失和损坏，因此任何一种密码设备应当具有密钥恢复的措施。从备份或存档中获取密钥的过程称为密钥恢复。若密钥丧失但未被泄露，就可以用安全方式从密钥备份中恢复。

密钥恢复措施需要考虑恢复密钥的效率问题，能在故障发生后及时恢复密钥。

密钥的更新

以下情况需要进行更新：

- 密钥有效期结束；
- 已知或怀疑密钥已泄漏；
- 通信成员中有人提出更新密钥。

更新密钥应不影响信息系统的正常使用，密钥注入必须在安全环境下进行并避免外漏。现用密钥和新密钥同时存在时应处于同等的安全保护水平下。更换下来的密钥一般情况下应避免再次使用，除将用于归档的密钥及时采取有效的保护措施以外应及时进行销毁处理。密钥更新可以通过再生密钥取代原有密钥的方式来实现。

密钥的存档/撤销/销毁

- 密钥存档：当密钥不再正常时，需要对其进行存档，以便在某种情况下特别需要时（如解决争议）能够对其进行检索。存档是指对过了有效期的密钥进行长期的离线保存，密钥的后运行阶段工作。
- 密钥撤销：若密钥丢失或在密钥过期之前，需要将它从正常使用的集合中删除。
- 密钥销毁：对于不再需要保留密钥及其相关联的内容，将清除所有与其相关的痕迹。

密钥安全审计

密钥管理中的安全审计是对在密钥的生存期中对密钥进行的各种操作及相关事件进行记录，以便及时发现问题，在事故发生后跟踪事故线索，追究事故责任。

密码安全审计记录应包括：

- 实施密钥管理和操作的人员、时间；
- 对密钥管理和操作的内容；
- 存放密钥的载体及标志；
- 可能泄露密钥的行为及涉及密钥安全的事件。

密钥的安全审计记录不应包含密钥本身，但可以包含其校验值。安全审计记录应有防止非授权修改和销毁的安全措施。

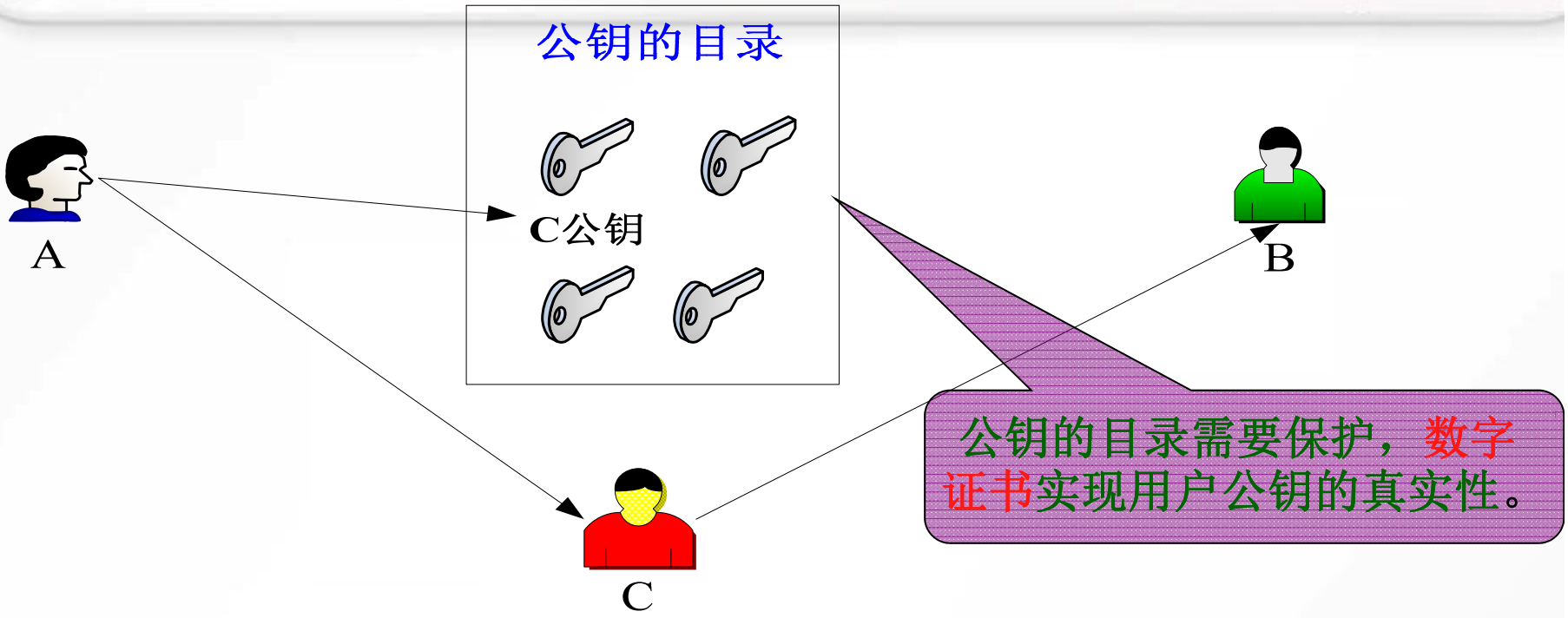
公钥密钥管理的简介

- 密码体制的不同，密钥的管理方法也不同；

  公钥密码的密钥管理与对称密码的密钥管理大不相同。

- 对称密码其实就一个密钥(即已知一个密钥可推出另一个密钥)，因此，密钥的秘密性、真实性、完整性都必须保护；
- 公钥密码有两个密钥，公钥和私钥是不同的，而且已知公钥在计算上不能求出私钥，因此，公钥的秘密性不用确保，但其真实性、完整性都必须严格保护；
- 公钥密码体制的私钥的秘密性、真实性、完整性都必须保护；

公钥存在的安全问题(中间人攻击)



1.C将公共目录中B的公钥替换成自己的公钥。
2.A将他认为的B的公钥提取出来，而实际上那是C的公钥。
3.C现在可以读取A送给B的加密信息。
4.C将A的信息解密并阅读，然后他又用真实的B的公钥加密该信息并将加密结果发送给B。

数字证书的引言

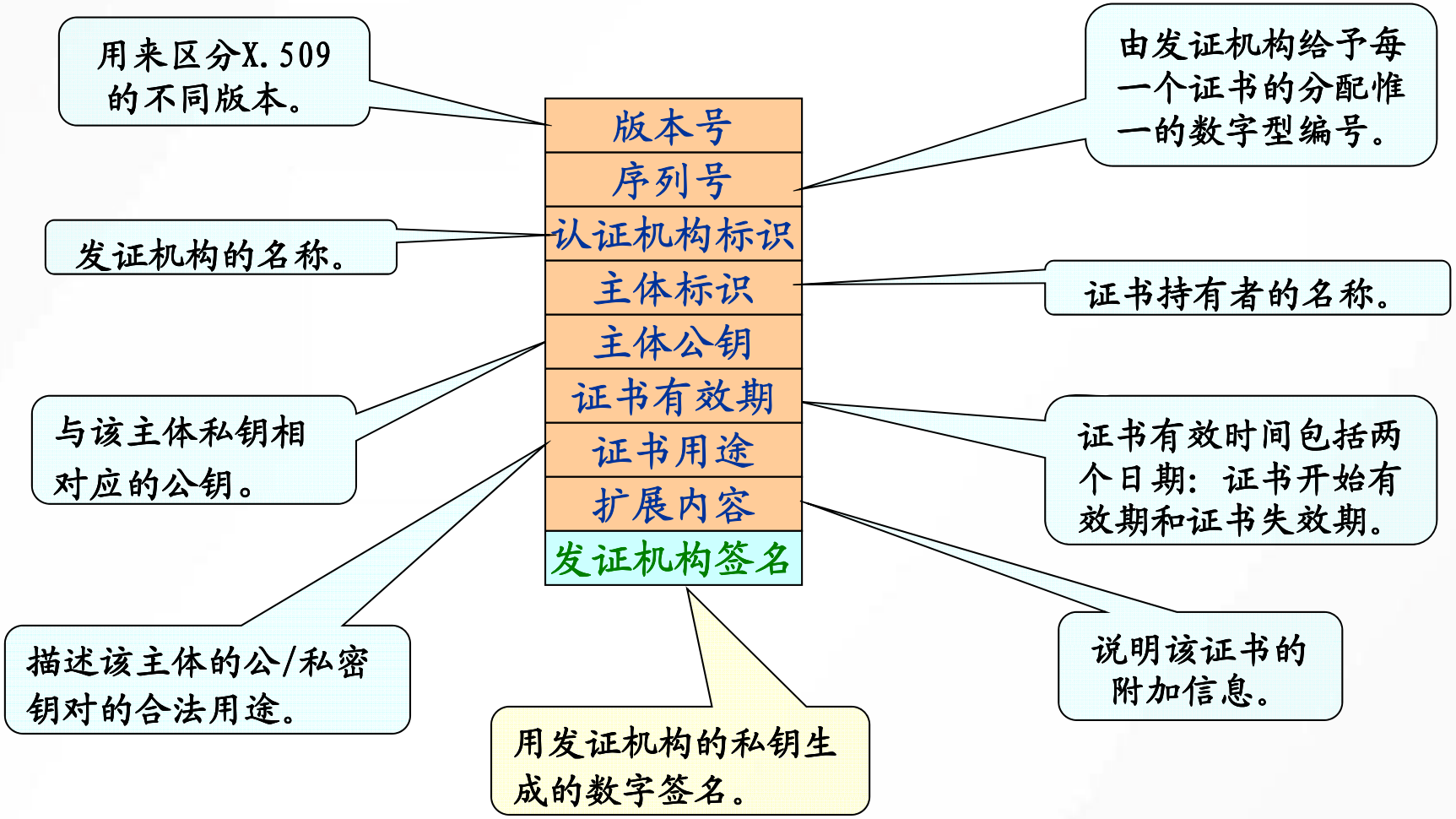
证书类似现实生活中的个人身份证。身份证将个人的身份信息(姓名、出生年月日、地址和其他信息)同个人的可识别特征(照片或指纹)绑定在一起。个人身份证是由国家权威机关(公安部)签发的，该证件的有效性和合法性是由权威机关的签名或签章保障的，因此身份证可以用来验证持有者的合法身份的信息，称为验证身份鉴定。

注:公钥算法的一个最大问题就是确认获得对方公钥的身份。

数字证书的概述

数字证书也称为公钥证书，是将证书持有者的身份信息和其所拥有的公钥进行绑定的文件。证书文件还包含签发该证书的权威机构认证中心CA对该证书的签名。通过签名保障了证书的合法性和有效性。证书包含的持有者公钥和相关信息的真实性和完整性也是通过CA的签名来保障的。这使得证书发布依赖于对证书本身的信任，也就是说证书提供了基本的信任机制。证书（和相关的私钥）可以提供诸如身份认证、完整性、机密性和不可否认性等安全服务。证书中的公钥可用于加密数据或验证对应私钥的签名。

数字证书的内容


版本号
序列号
认证机构标识
主体标识
主体公钥
证书有效期
证书用途
扩展内容
发证机构签名
用来区分X.509的不同版本。
由发证机构给予每一个证书的分配惟一的数字型编号。
发证机构的名称。
证书持有者的名称。
与该主体私钥相对应的公钥。
证书有效时间包括两个日期：证书开始有效期和证书失效期。
描述该主体的公/私密钥对的合法用途。
说明该证书的附加信息。
用发证机构的私钥生成的数字签名。

数字证书的安全性

- 证书是公开的，可复制的。
- 任何具有**CA**公钥（根证书/**CA**证书，自签名证书）的用户都可以验证证书有效性
- 除了**CA**以外，任何人都无法伪造、修改证书。
- 证书的安全性依赖于**CA**的私钥。

数字证书的理解

- 数字证书(Digital ID)又叫”网络身份证”、”数字身份证”；
- 持证的主体可以是人、设备、组织机构或其他主体；
- 包含公开密钥拥有者以及公开密钥相关信息的一种电子文件，能以明文的形式进行存储和分配；
- 可以用来证明数字证书持有者的真实身份；
- 由认证中心发放并经认证中心数字签名的，是PKI体系中最基本的元素；
- 证书是一个机构颁发给一个安全个体的证明，所以证书的权威性取决于该机构的权威性。

数字证书的分类

从证书的基本用途来看：

- 签名证书

  签名证书主要用于对用户信息进行签名，以保证信息的不可否认性；（私钥不需备份）

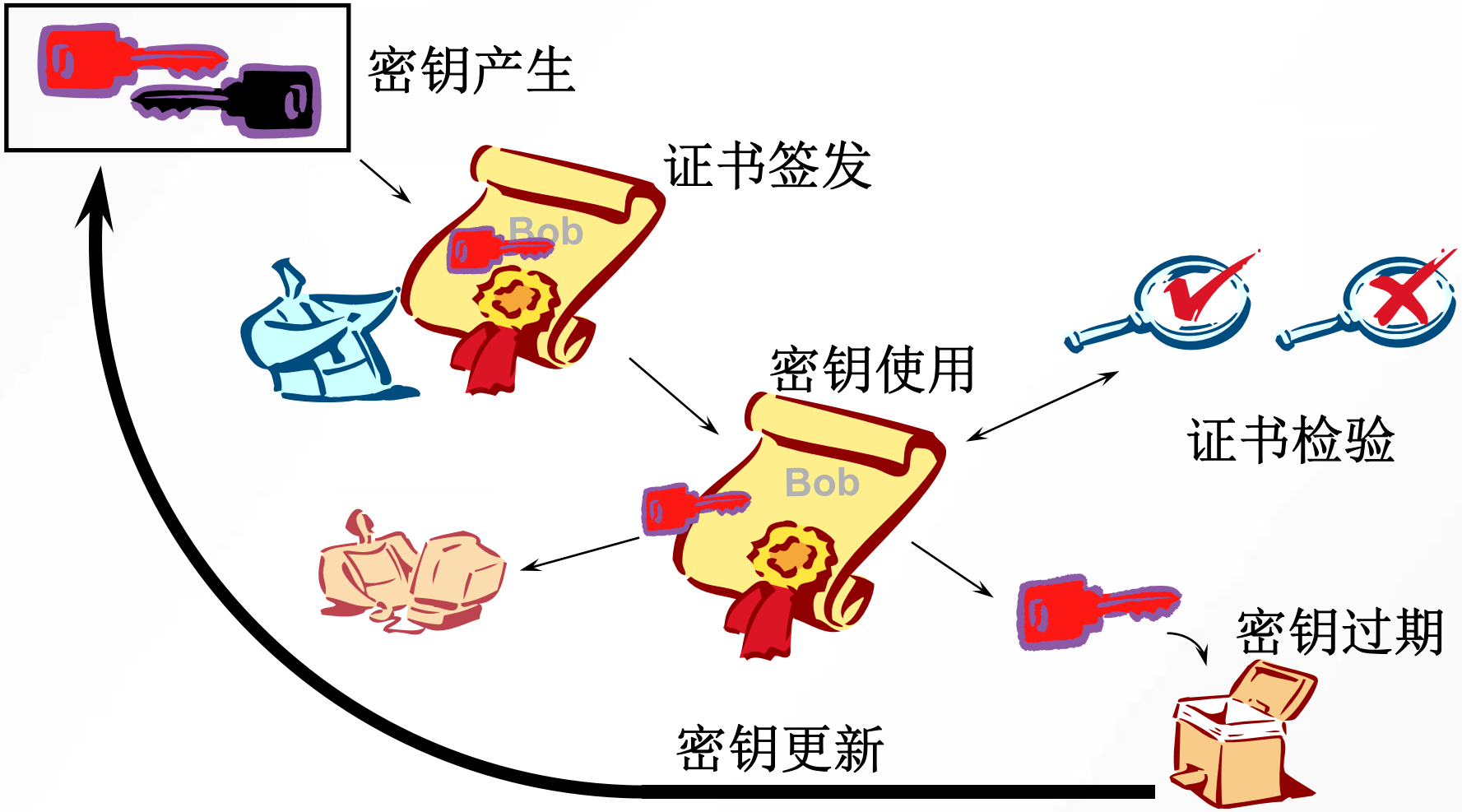
- 加密证书

  加密证书主要用于对用户传送信息进行加密，以保证信息的真实性和完整性。（私钥需要备份）

证书管理

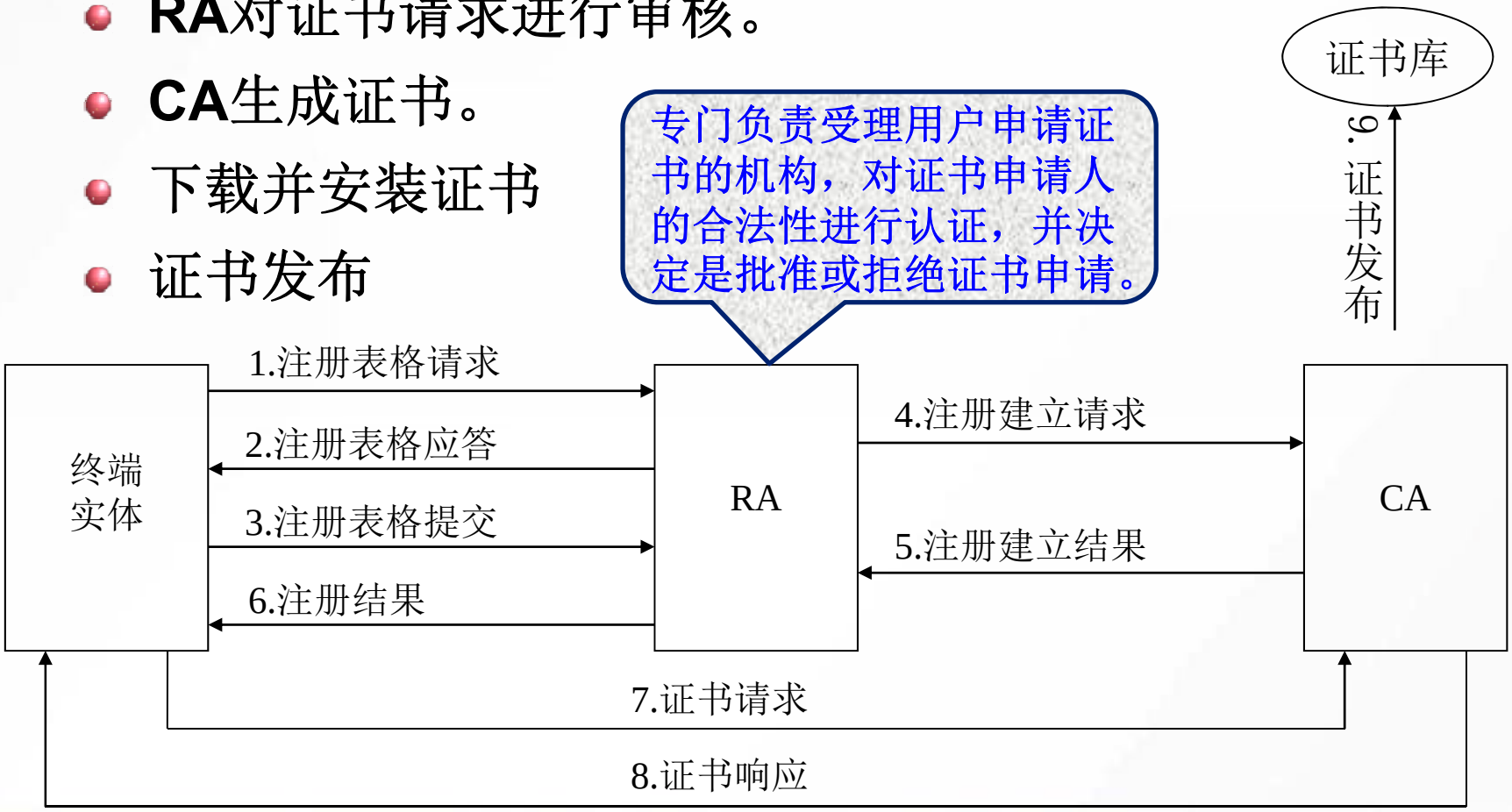
- 证书注册
- 证书更新
- 证书存放
- 证书撤销
- 证书验证
- 证书状态查询

密钥生命周期



证书注册

- 申请人提交证书请求。
- **RA**对证书请求进行审核。
- **CA**生成证书。
- 下载并安装证书
- 证书发布



证书的更新

- 更新原因
  - 证书过期；
  - 一些属性的改变；
  - 证书的公钥对应的私钥泄露。
- 最终实体证书更新

  一般发放新证书。
- CA证书更新

  产生新CA证书和新用旧证书(用新证书的私钥签名)。

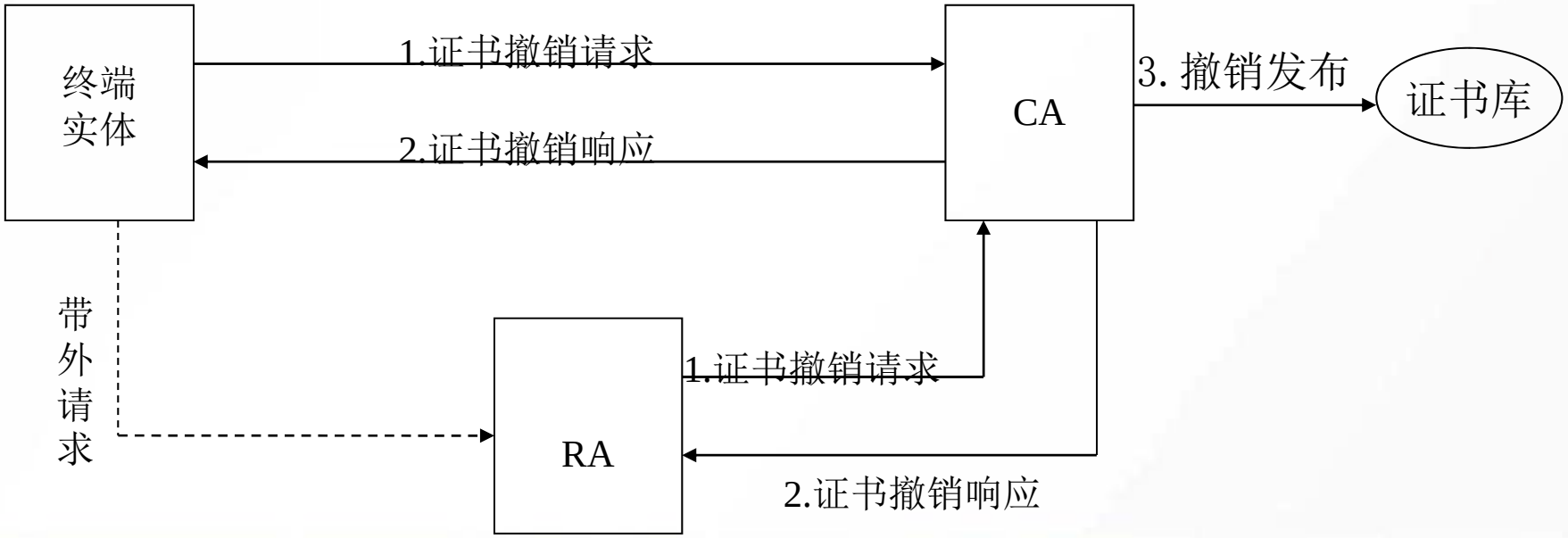
  保证实体的旧证书仍能使用，直到所有旧证书都过期，取消新用旧证书；

证书存放

- 使用IC卡存放
- 直接存放在磁盘或自己的终端上
- USB Key
- 证书库

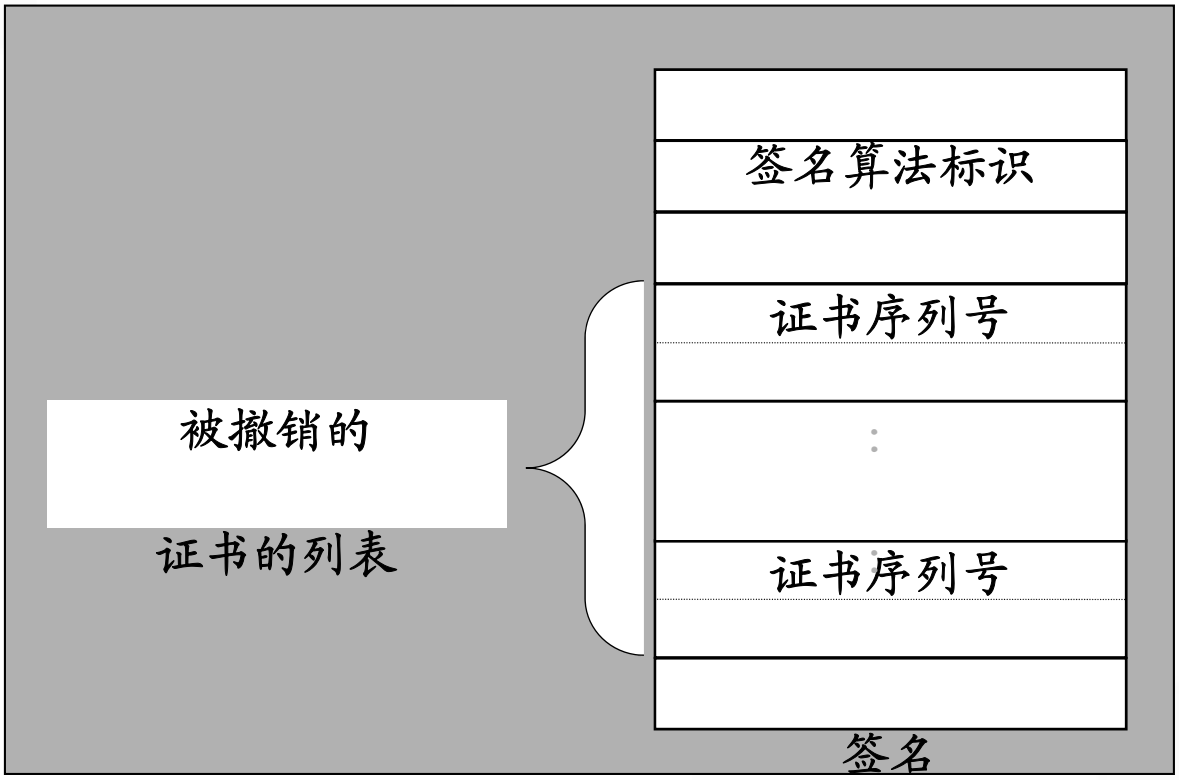
证书撤销

- 当条件（雇佣关系结束、证书中信息修改等）要求证书的有效期在证书结束日期之前终止，或者要求用户与私钥分离时（私钥可能以某种方式泄露），证书被撤销。



证书撤销列表

用户在使用一个证书之前，必须检查证书是否已被撤销



证书撤销列表（CRL）会无限增加吗？

证书验证

- 使用CA证书验证终端实体证书有效性。
- 检查证书的有效期，确保该证书是否有效。
- 检查该证书的预期用途是否符合CA在该证书中指定的所有策略限制。
- 在证书撤销列表（CRL）中查询确认该证书是否被CA撤销。

证书状态查询

- 定期下载证书撤销列表（**CRL**）。
- 在线证书状态协议**OCSP**（**Online Certificate Status Protocol**），其目的为了克服基于**CRL**的撤销方案的局限性，为证书状态查询提供即时的最新响应。**OCSP**使用证书序列号、**CA**名称和公开密钥的散列值作为关键字查询目标的证书。

PKI信任模型

- 单一模型CA
- 树状模型CA
- 对等模型CA

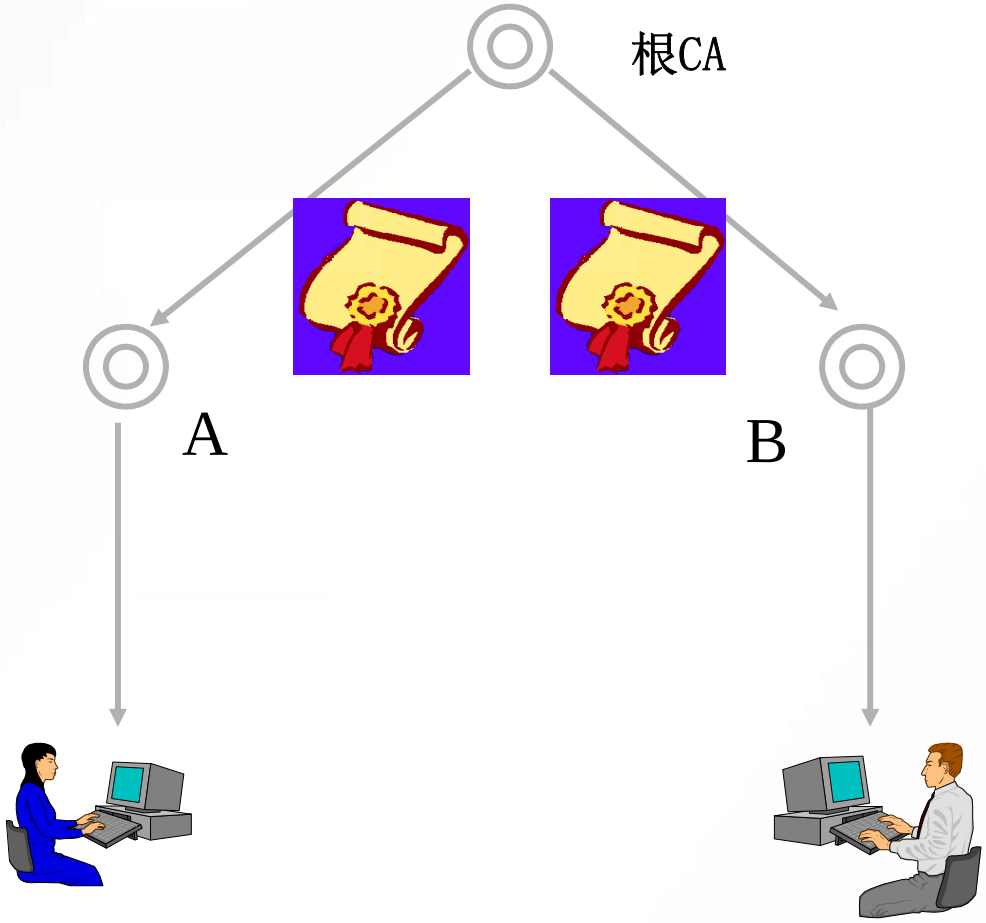
单一模型CA

比较理想建立单一的认证机构（CA），由它管理所有证书：

- 特殊的应用领域有着特殊的安全需求（安全策略不同）。
- 证书颁发和撤销难于直接控制。
- 证书库难于维护，难于满足实际性能的需求。
- 根私钥的维护更加困难。

多CA带来问题是如何建立CA之间的信任关系

树状模型CA



A，B均用根CA所发证书完成初始化

层次结构CA中证书的验证

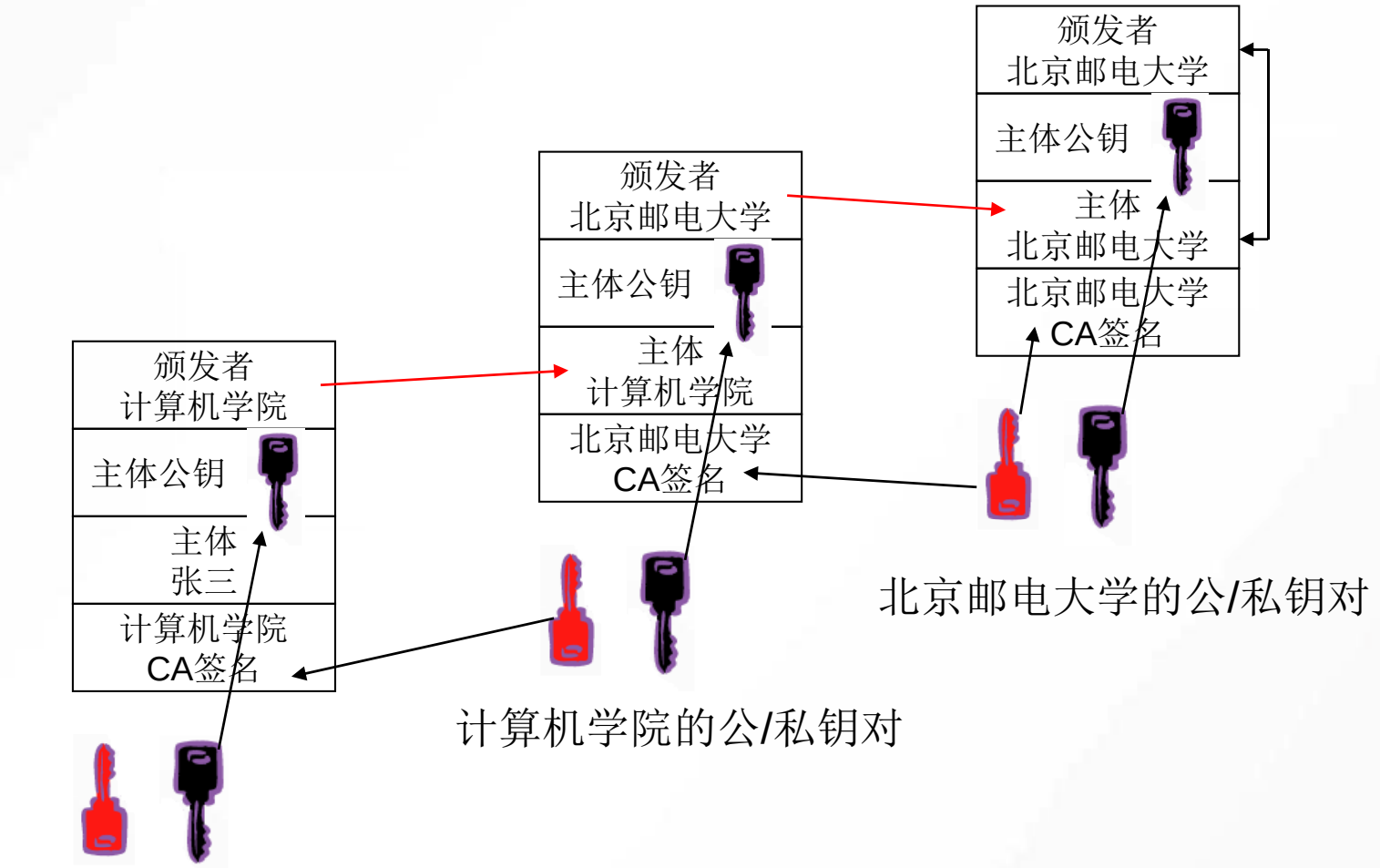
- 假设个体A看到B的一个证书；
- B的证书中含有签发该证书的CA的信息；
- 沿着层次树往上找，可以构成一条证书链，直到根证书；
- 验证过程：

➢ 沿相反的方向，从根证书开始，依次往下验证每一个证书中的签名。其中，根证书是自签名的，用它自己的公钥进行验证；

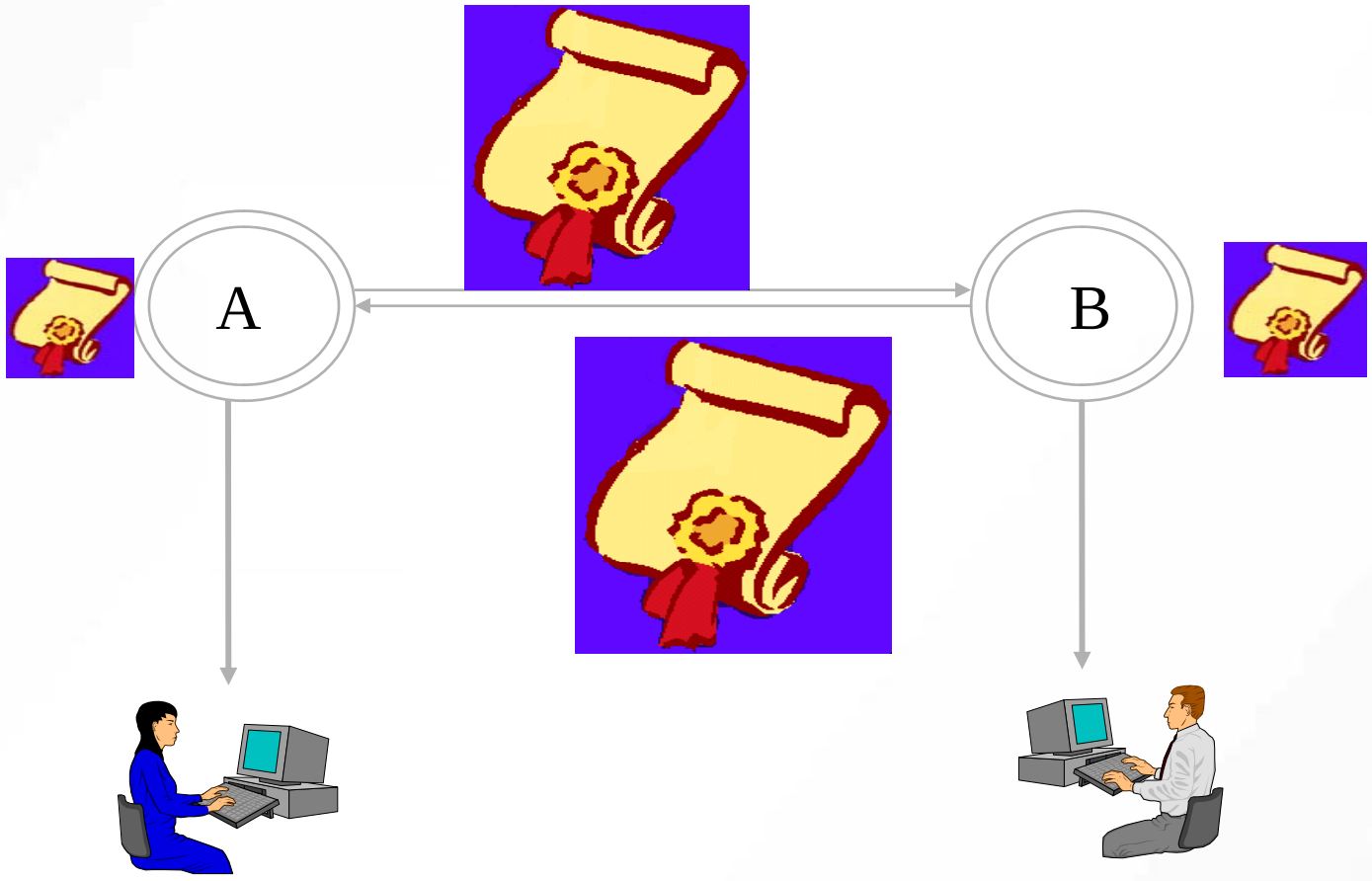
➢ 一直到验证B的证书中的签名；

➢ 如果所有的签名验证都通过，则A可以确定所有的证书都是正确的，如果他信任根CA，则他可以相信B的证书和公钥；

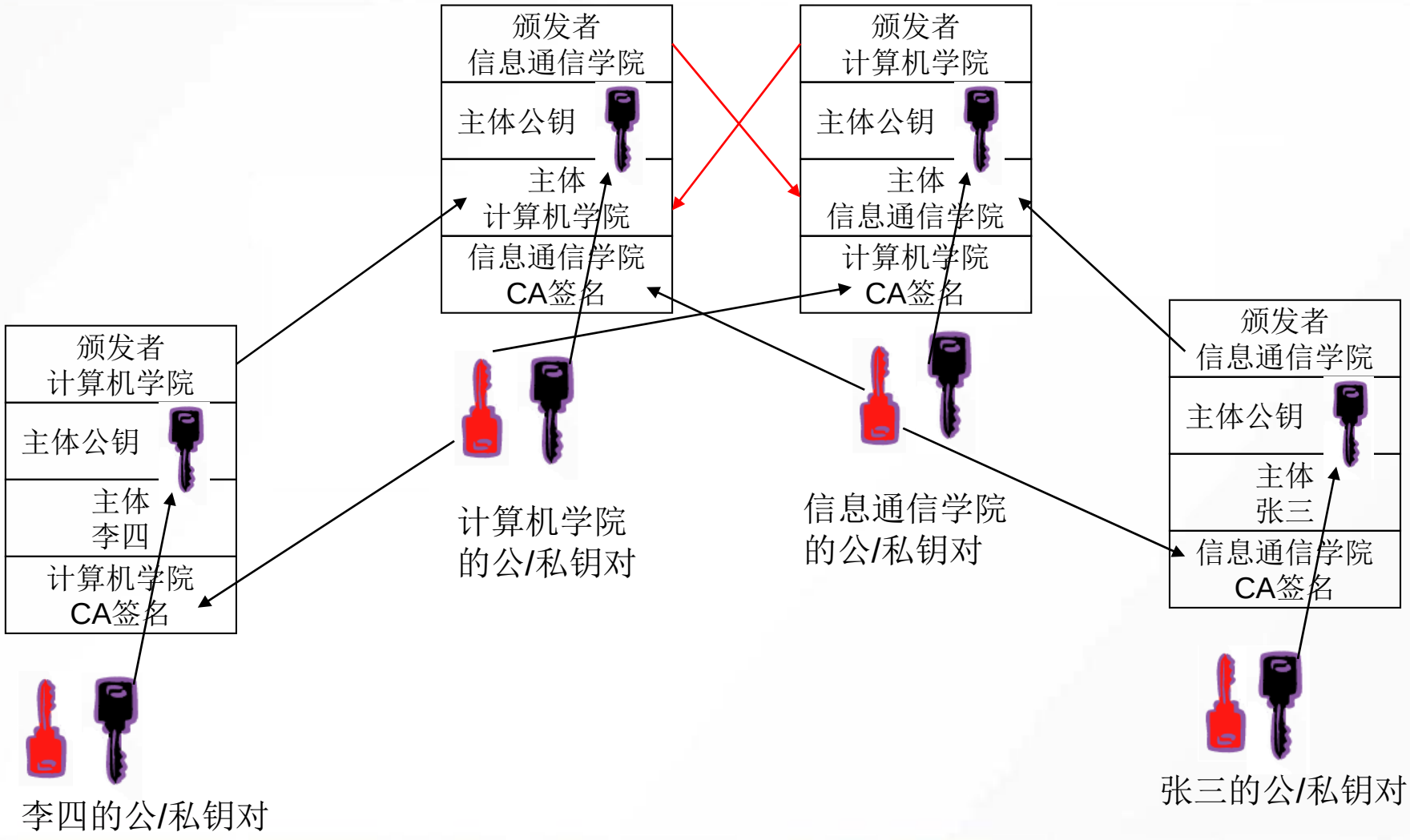
树状模型CA（续）


颁发者
北京邮电大学
主体公钥
主体
北京邮电大学
北京邮电大学
CA签名
颁发者
北京邮电大学
主体公钥
主体
计算机学院
北京邮电大学
CA签名
颁发者
计算机学院
主体公钥
主体
张三
计算机学院
CA签名
北京邮电大学的公/私钥对
计算机学院的公/私钥对
用户的公/私钥对

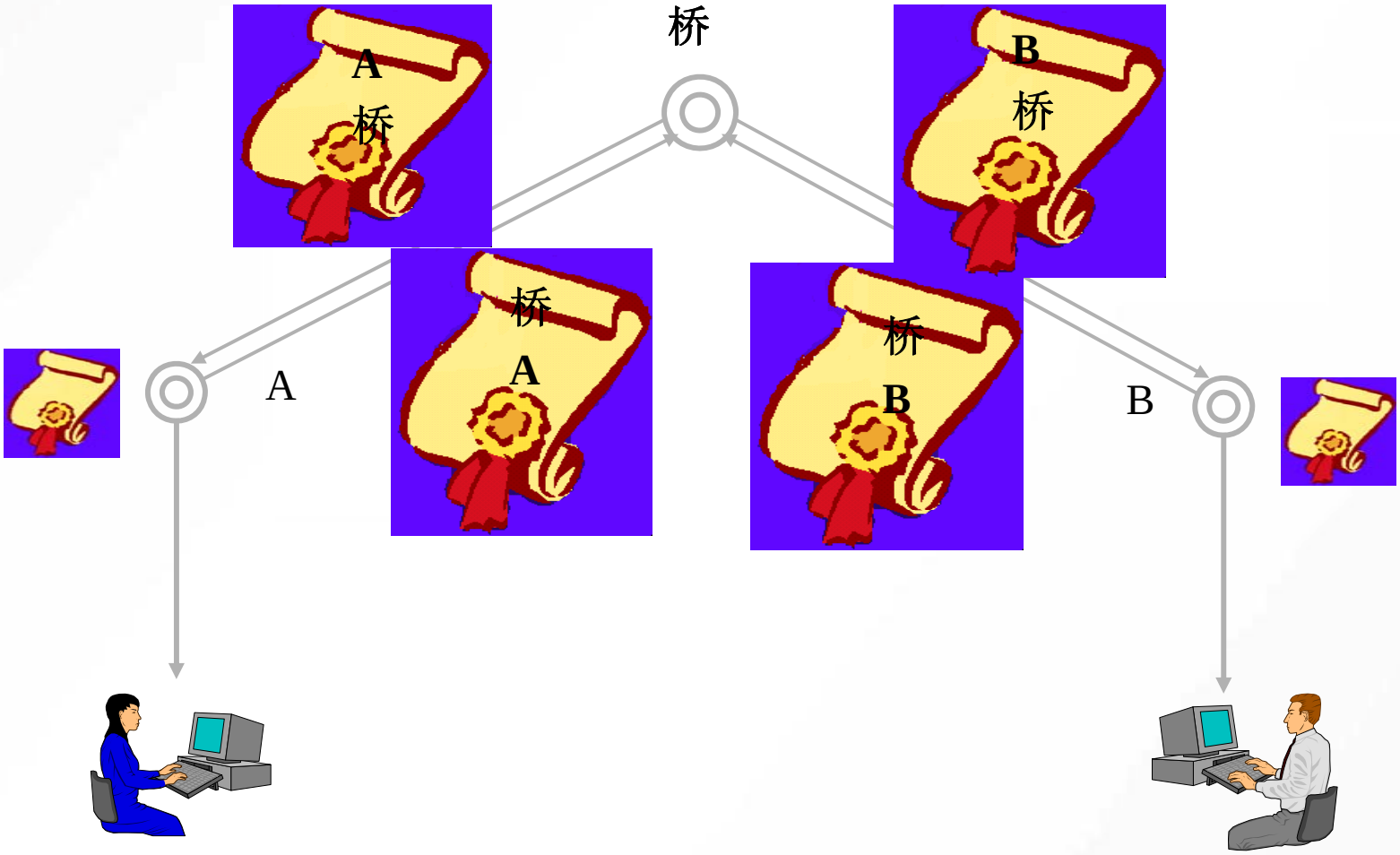
对等模型CA



对等模型CA（续）


颁发者
信息通信学院
主体公钥
主体
计算机学院
信息通信学院
CA签名
颁发者
计算机学院
主体公钥
主体
信息通信学院
计算机学院
CA签名
颁发者
计算机学院
主体公钥
主体
李四
计算机学院
CA签名
计算机学院
的公/私钥对
信息通信学院
的公/私钥对
颁发者
信息通信学院
主体公钥
主体
张三
信息通信学院
CA签名
李四的公/私钥对
张三的公/私钥对

网状模型CA



密钥分配的简介

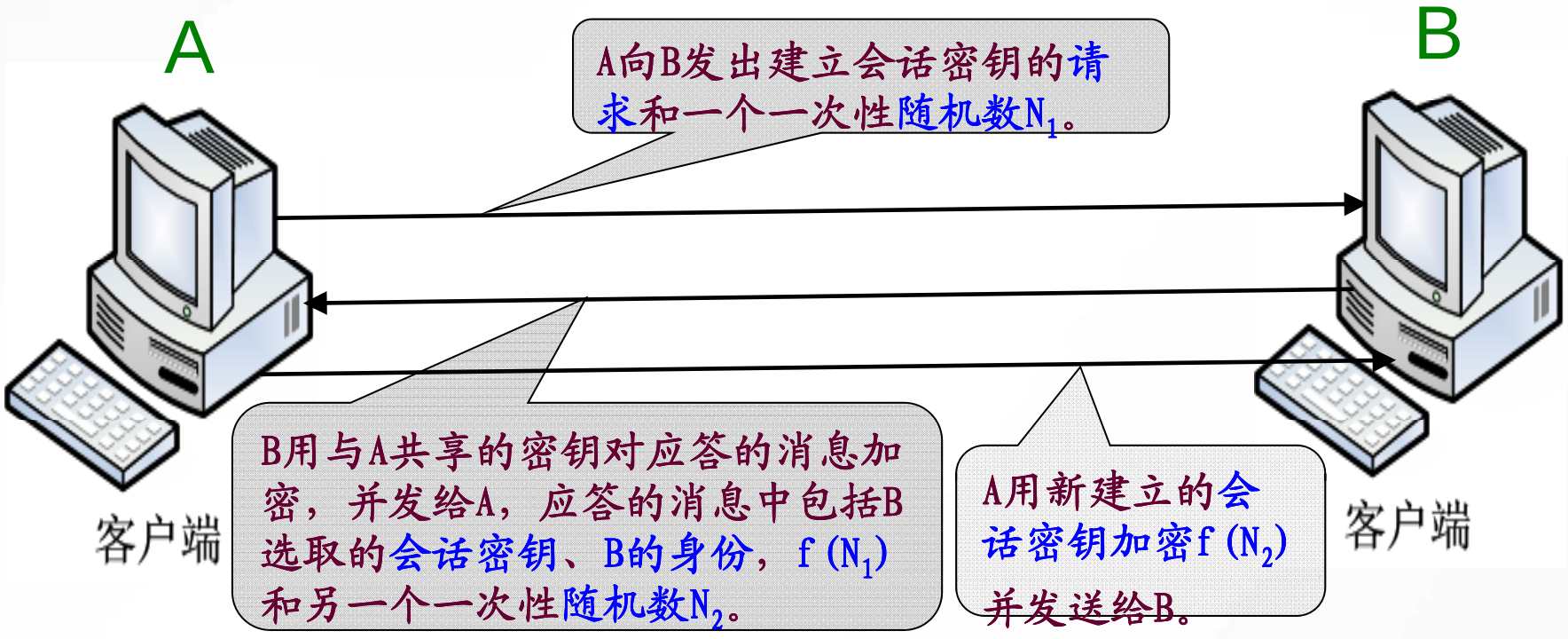
- 密钥分配就是一种机制，通过这种机制，通信双方中的一方或密钥分配中心选取一个秘密密钥，然后将其传送给通信双方中的另一方。 目的
- 密钥分配技术是在不让其他人（除密钥分配中心）看到密钥的情况下将一个密钥传递给希望交换数据的双方的方法。 安全
- 为防止攻击者得到密钥，必须时常更新密钥，密码系统的强度依赖于密钥分配技术。 重要

密钥分配的基本方法

- *密钥由A选取并通过物理手段发送给B。*
- *密钥由第三方选取并通过物理手段发送给A和B。*
- 如果A、B事先已有一密钥，则其中一方选取新密钥后，用已有的密钥加密新密钥并发送给另一方。 无中心
- 如果A和B与第三方C分别有一保密信道，则C为A、B选取新密钥后，其中一方把新密钥安全传送给另一方。 有中心
- 如果A、B事先有自己的私钥，则其中一方选取新密钥，利用PKI技术实现新密钥安全传送。 基于证书

无中心的密钥分配模式

前提条件：通信双方有共享密钥。



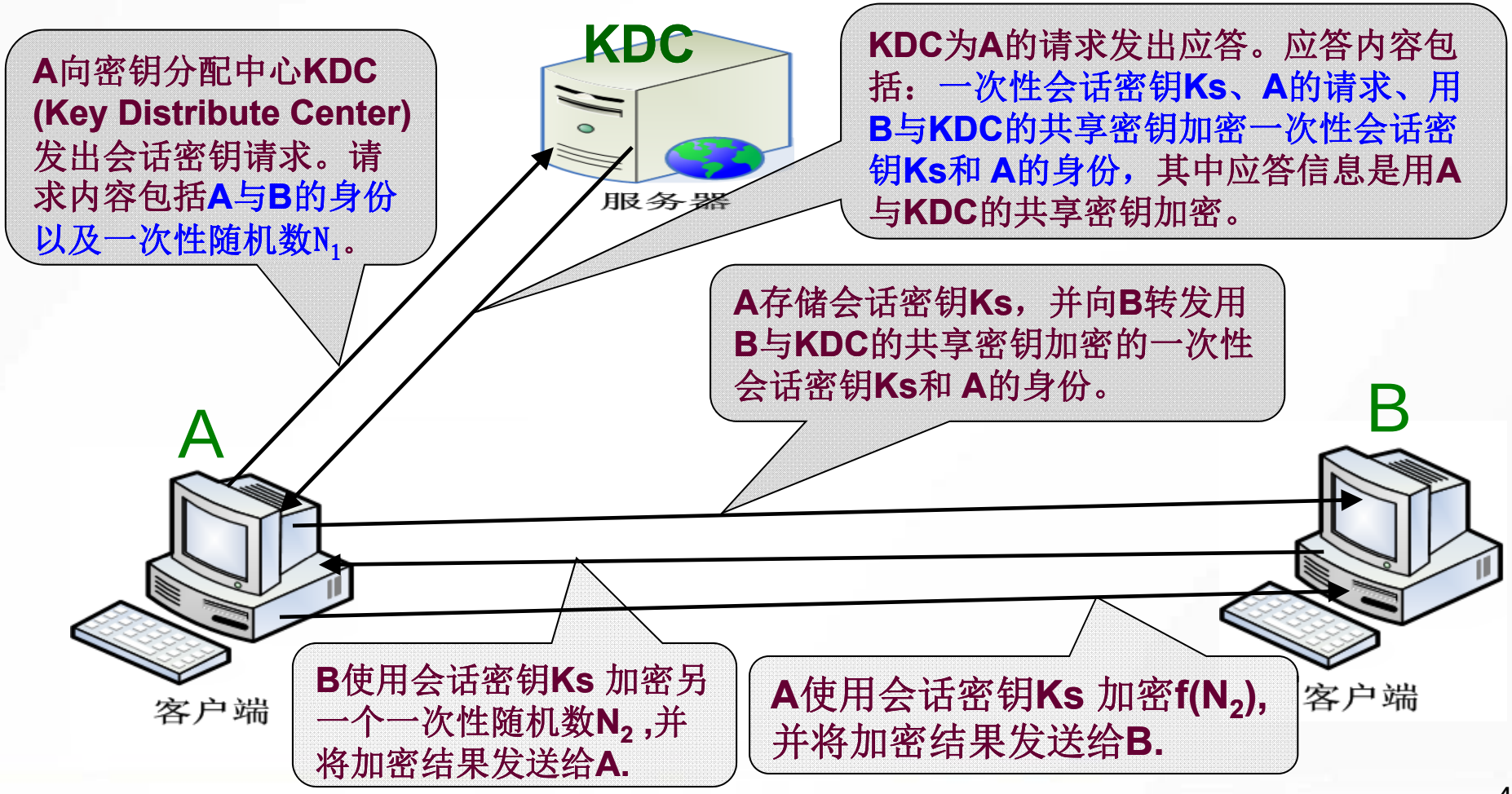
问题：为什么分配密钥需要随机数？

无中心密钥分配方法的分析

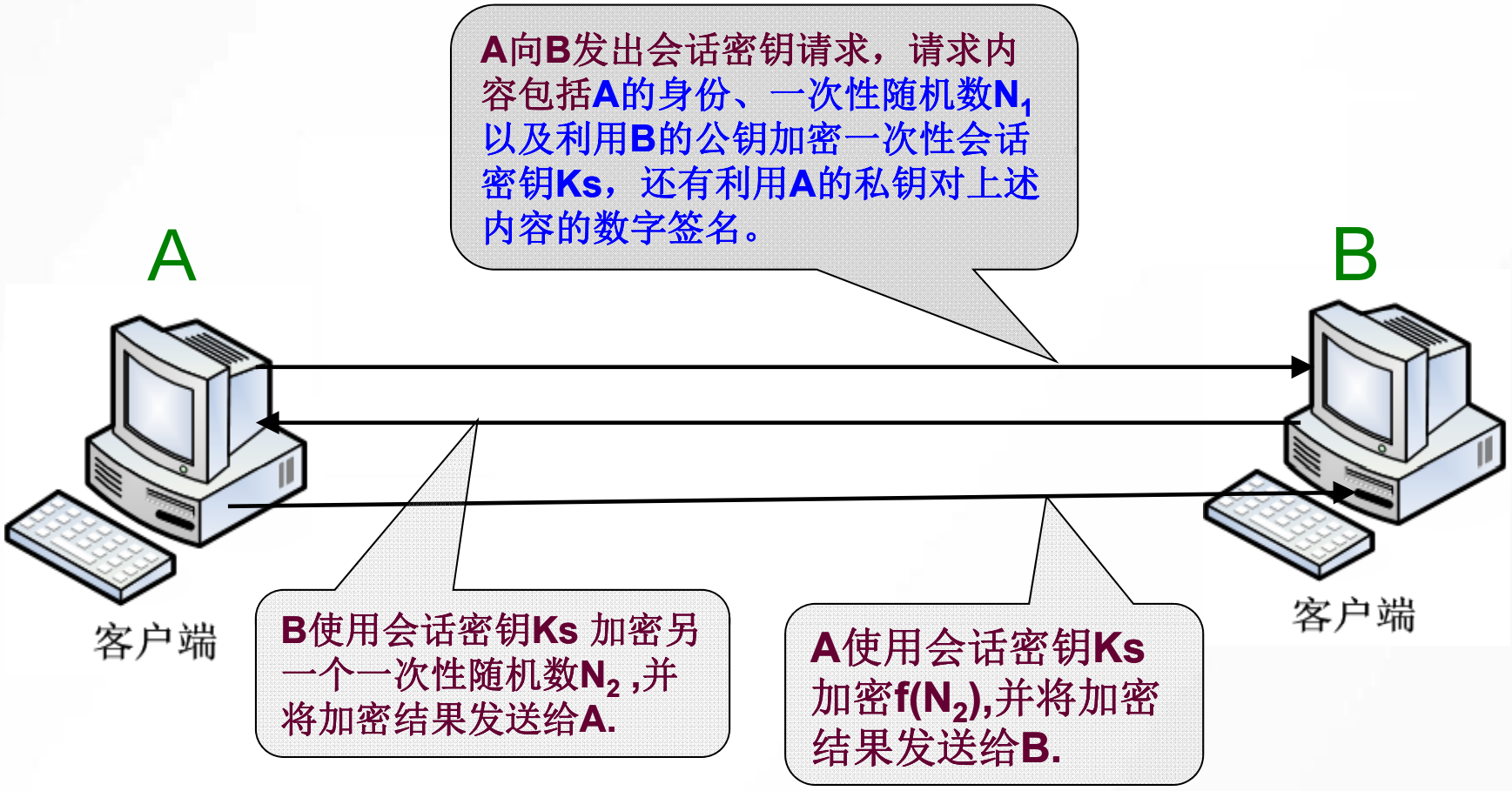
- 如果所有用户都要求支持加密服务，则任意一对希望通信的用户都必须有一共享密钥。如果有**n**个用户，每个用户需要存储**n-1**个密钥，这对每个用户都是一个沉重的负担，另，密钥数目为**n(n-1)/2**，密钥分配也是一个难题。
- *建立一个负责为用户分配密钥的密钥分配中心，这时每一用户必须和密钥分配中心有一个共享密钥，称为主密钥。通过主密钥分配给一对用户的密钥称为会话密钥，用于这一对用户之间的保密通信。通信完成后，会话密钥即被销毁。如上所述，如果用户数为**n**，则会话密钥数为**n(n-1)/2**，但主密钥数却只需**n**个。*

有中心的密钥分配模式

前提条件：密钥分配中心与每个用户之间有共享密钥。



基于公钥证书实现密钥分配



密钥协商

- 密钥协商是保密通信双方（或更多方）通过公开信道的通信来共同形成秘密密钥的过程。一个密钥协商方案中，密钥的值是某个函数值，其输入量由两个成员（或更多方）提供。
- 密钥协商的结果是：参与协商的双方（或更多方）都将得到相同的密钥，同时，所得到的密钥对于其他任何方都是不可知的。

Diffie-Hellman密钥交换方案

设p是一个大素数，$g \in Z_p$是模p本原元，p和g公开，所有用户均可获取，并可为所有用户所共有。

⑴用户A随机选取一个大数a，$0 \le a \le p-2$。

⑵用户A计算$K_a = g^a \pmod p$，并将结果传送给用户B。

⑶用户B随机选取一个大数b，$0 \le b \le p-2$。

⑷用户B计算$K_b = g^b \pmod p$，并将结果传送给用户A。

⑸用户A计算$K = (K_b)^a \pmod p$。

⑹用户B计算$K = (K_a)^b \pmod p$。

用户A和用户B各自计算生成共同的会话密钥K。

这是因为：$K = (K_b)^a = (g^b)^a = g^{ab} = (g^a)^b = (K_a)^b$。

Diffie-Hellman密钥交换方案(扩展)

下面以 A、B 和 C 三方一起产生秘密密钥为例。

(1) A 选取一个大随机整数 x，并且发送 $X \equiv g^x \bmod p$ 给 B。

(2) B 选取一个大随机整数 y，并且发送 $Y \equiv g^y \bmod p$ 给 C。

(3) C 选取一个大随机整数 z，并且发送 $Z \equiv g^z \bmod p$ 给 A。

(4) A 发送 $Z' \equiv Z^x \bmod p$ 给 B。

(5) B 发送 $X' \equiv X^y \bmod p$ 给 C。

(6) C 发送 $Y' \equiv Y^z \bmod p$ 给 A。

(7) A 计算 $k \equiv Y'^x \bmod p$。

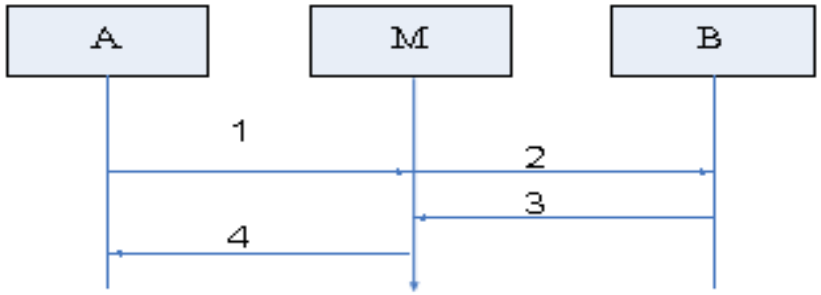
(8) B 计算 $k \equiv Z'^y \bmod p$。

(9) C 计算 $k \equiv X'^z \bmod p$。

显然，秘密密钥 k 等于 $g^{xyz} \bmod p$，除了他们三人外，没有别的人能计算出 k 值。

中间人攻击

Diffie-Hellman 密钥交换协议不包含通信双方的身份认证过程，所以，处于通信双方 A 和 B 通信中间的攻击者能够截获并替换他们之间的密钥协商交互的消息，从而监听到他们的通信内容，这种攻击被称为中间人攻击。



1. A 选择 $a \in_u [1, p-1)$，计算 $g_a \equiv g^a \pmod p$，发送 g_a 给 M（“B”）；

2. M（“A”）对某个 $m \in [1, p-1)$，计算 $g_m \equiv g^m \pmod p$，发送 g_m 给 B；

3. B 选择 $b \in_u [1, p-1)$，计算 $g_b \equiv g^b \pmod p$，发送 g_b 给 M（“A”）；

4. M（“B”）向 A 发送 g_m；

5. A 计算 $k_1 \equiv {g_m}^a \pmod p$；

6. B 计算 $k_2 \equiv {g_m}^b \pmod p$；

上面过程完成之后，攻击者 M 可以计算出密钥 k_1 和 k_2，并用这两个密钥就可以监听 A 和 B 之间的秘密通信。A 用 $g^{am} \bmod p$ 加密信息发送给 B，M 截取信息的密文并用 k_1 解密得到信息，随后用密钥 $g^{bm} \bmod p$ 再次加密信息并传送给 B。反之亦然。

DH密钥协商协议的改进

该协议基于公钥基础设施引入了数字签名算法，假定存在可信中心 CA，其签名算法用 $Sign$ 表示，与之对应的签名验证算法用 Ver 表示。域中的每个用户可以事先向 CA 注册并申请一个公钥证书，则改进的协议描述如下：

设 p 是一个大素数，$g \in Z_p$ 是模 p 的一个本原元，p 和 g 公开。

(1) A 随机选取 x，$0 \le x \le p-2$；计算 $g_a \equiv g^x (\mathrm{mod}\, p)$，并将结果传送给用户 B。

(2) B 随机选取 y，$0 \le y \le p-2$；计算 $g_b \equiv g^y (\mathrm{mod}\, p)$。然后计算：$S_B = Sign_A(g_a, g_b)$，用户 B 将 $(C(B), g_b, S_B)$ 传送给用户 A。

(3)用户 A 先验证 $C(B)$ 的有效性，然后验证 B 的签名 S_B 的有效性。确认 S_B 有效后，计算 $S_A = Sign_A(g_a, g_b)$，把自己的公钥证书以及签名 S_A 发给用户 B。最后，计算 $K \equiv g_b^x \bmod p$ 作为会话密钥。

(4)B 同样先验证 $C(A)$ 的有效性，然后验证 A 的签名 S_A 的有效性。确认 S_A 有效后，计算 $K \equiv g_b^x \bmod p$ 作为会话密钥。

其中：$C(A)$ 和 $C(B)$ 分别表示用户 A 和 B 的证书，$Sign_A$ 和 $Sign_B$ 分别表示利用 A 私钥和 B 私钥进行签名算法。

Blom方案

- TA公开一个素数，同时，每个用户U公开一个元素$r_U \in Zp$，不同用户之间的r_U必须不同。
- TA选择三个随机元素$a,b,c \in Zp$(不必不同)，且形成下列多项式：$f(x,y)=a+b(x+y)+cxy \pmod p$。
- 对每一个用户U，TA计算多项式：$g_U(x)=f(x, r_U) \pmod p$ $=a_U+b_Ux \pmod p$。其中：$a_U=a+br_U \pmod p$ 。$b_U= b+cr_U \pmod p$。同时，TA通过安全信道传送$g_U(x)$（传送多项式的系数）给用户U。
- 如果用户U和V想通信，那么，将他们将使用一个公共密钥：$K_{V,U}=K_{U,V}=f(r_U,r_V)=a+b(r_U+r_V)+cr_Ur_V \pmod p=g_U(r_V)=g_V(r_U)$。

Blom方案的分析

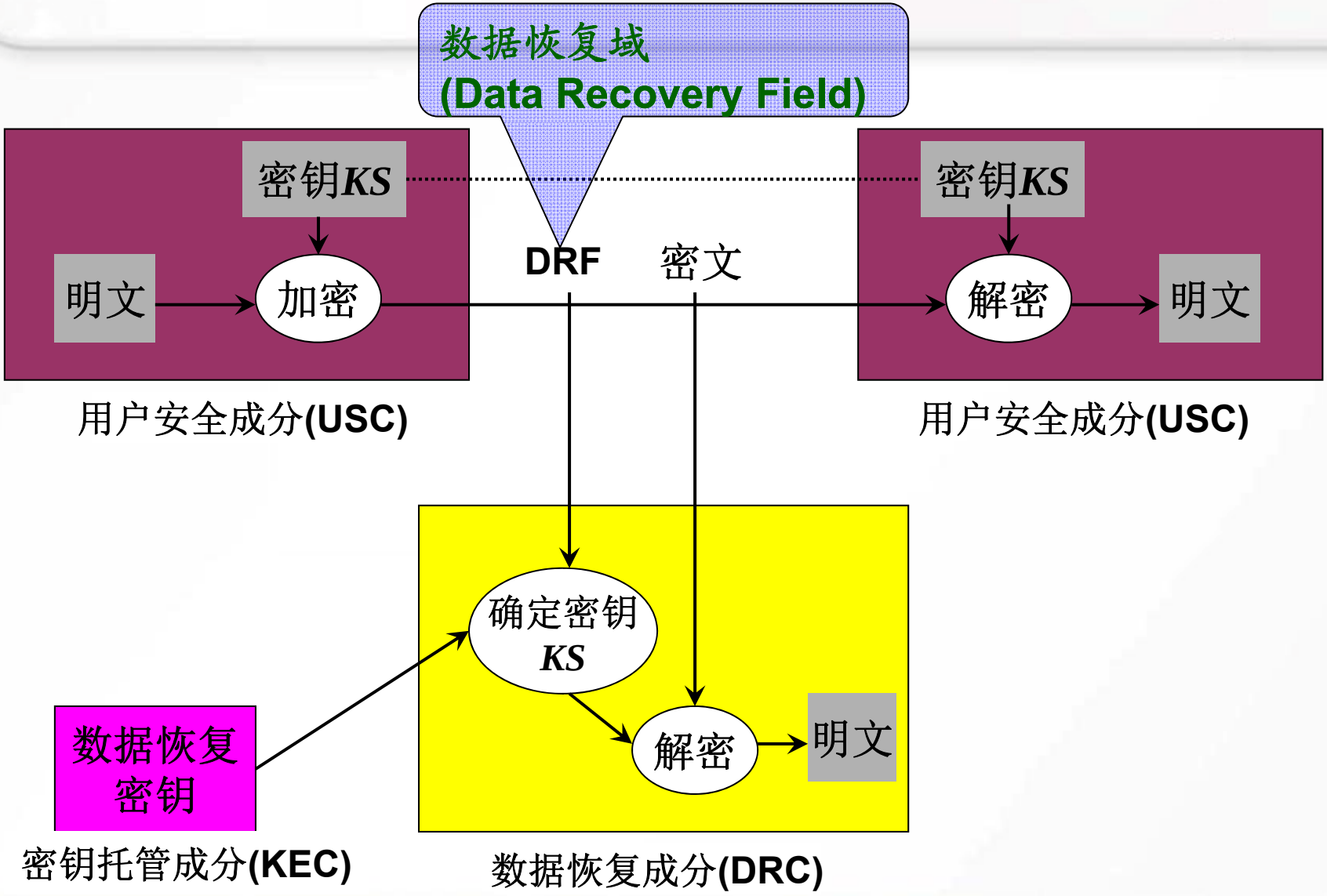
- 一个用户不能确定另外两个用户间的会话密钥。
- 若两个用户串通，Blom方案是不安全的。
- 可信中心知道每对用户的会话密钥。
- 每对用户间的会话密钥总是固定不变的。

密钥托管

密钥托管也称为托管加密，其目的是保证对个人没有绝对的隐私和绝对不可跟踪的匿名性，即在强加密中结合对突发事件的解密能力。其实现手段是把已加密的数据和数据恢复密钥联系起来，数据恢复密钥不必是直接解密的密钥，但由它可得解密密钥。

也就是说，密钥托管是指为公众和用户提供更好的安全通信同时，也允许授权者（包括政府保密部门、企业专门技术人员和用户等）为了国家、集团的利益，监听某些通信内容并解密相关密文。密钥托管也叫“密钥恢复”，或者理解为受信任的第三方的“数据恢复”和“特殊获取”等含义。

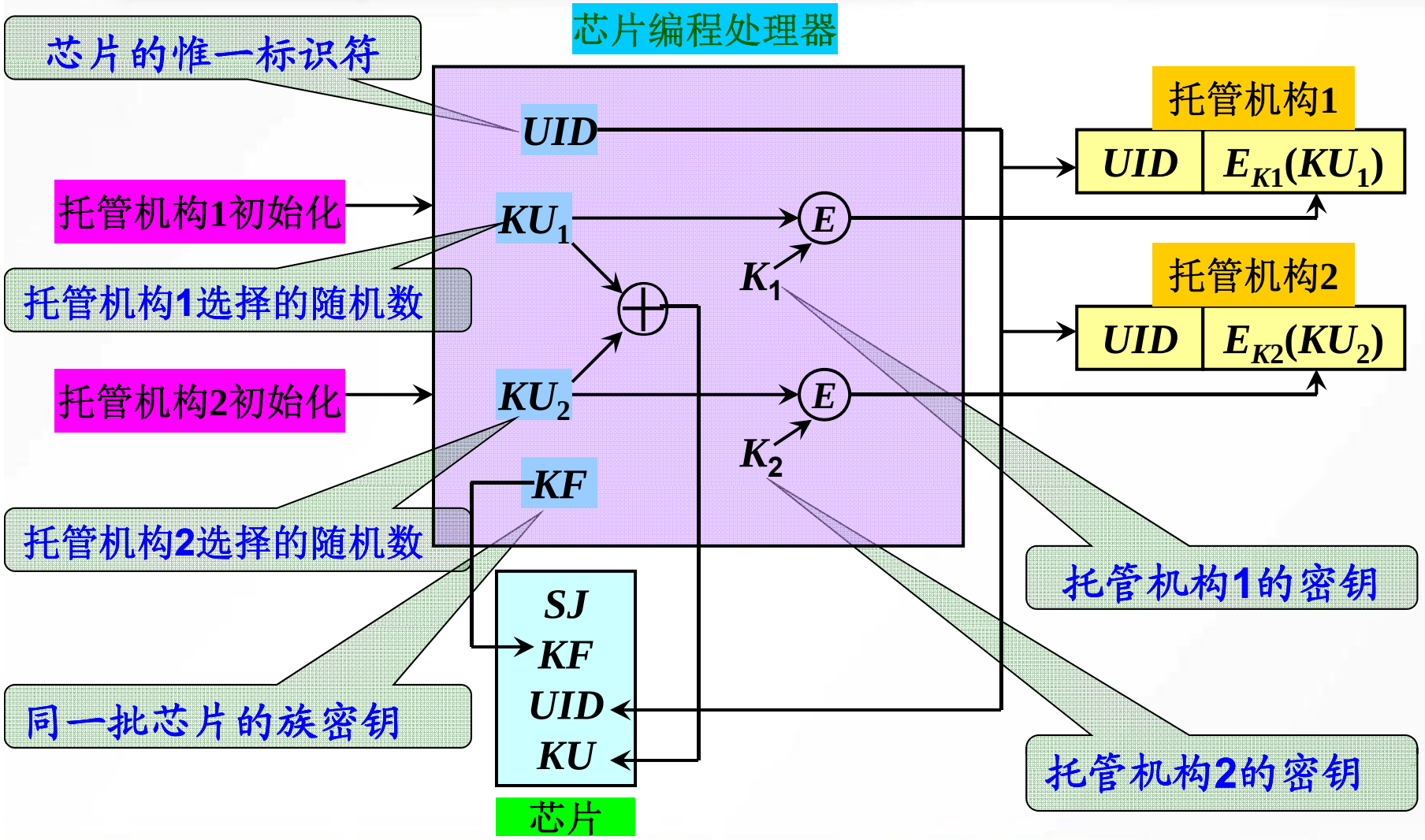
密钥托管密码体制的组成图



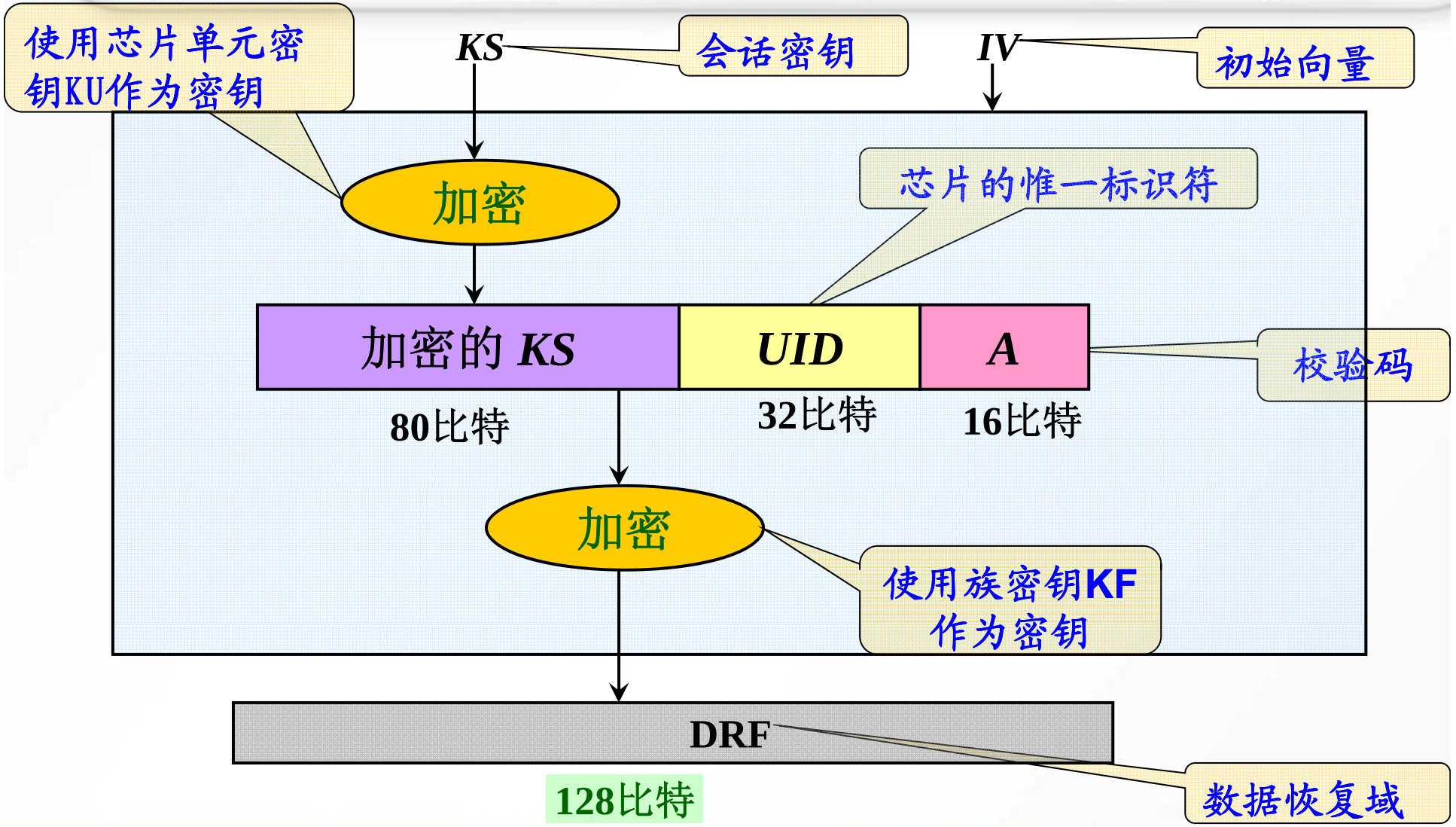
密钥托管的举例（Skipjack算法）

- 托管加密芯片的编程过程;
- 托管加密芯片加密;
- 托管加密芯片通信的存取;

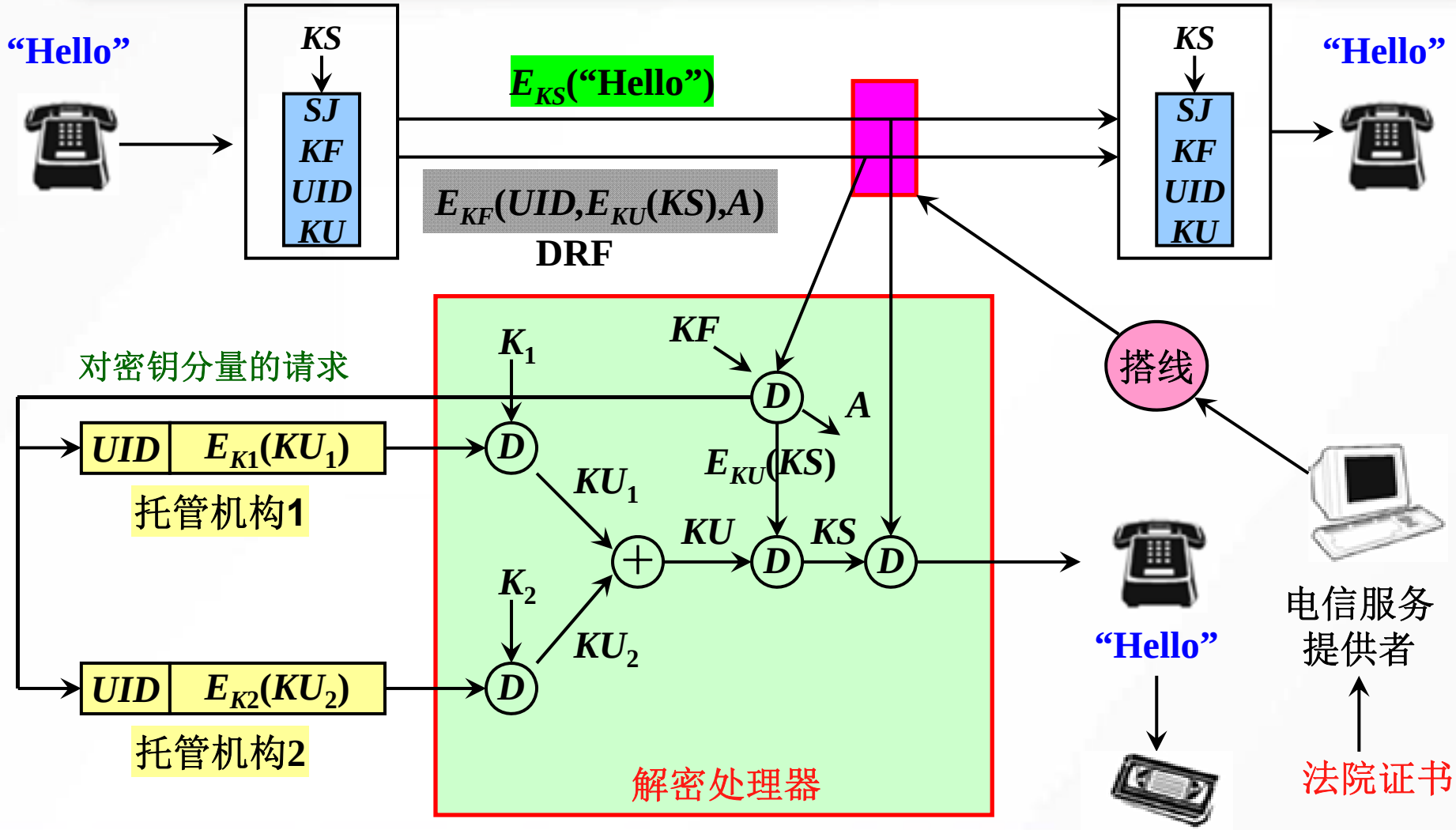
托管加密芯片的编程过程



托管加密芯片加密(生成数据恢复域)



托管加密芯片通信的监听



秘密分割门限方案

秘密s被分为n个部分，每个部分称为子密钥，由一个参与者持有，使得：

- 由t个或多于t个参与者所持有的部分信息可重构s。
- 由少于t个参与者所持有的部分信息则无法重构s。
- 称为(t, n)秘密分割门限方案，t称为门限值。
- 少于t个参与者所持有的部分信息得不到s的任何信息（并不比局外人猜s有优势）称该门限方案是完善的。

Shamir门限方案

- 有限域GF(q),q为大素数，$q \geqslant n+1$。秘密s是$GF(q)\setminus\{0\}$上均匀选取的随机数，表示为$s \in_R GF(q)\setminus\{0\}$. t-1个系数$a_1, a_2, \cdots a_{t-1}$选取$a_i \in_R GF(q)\setminus\{0\}$. 在GF(q)上构造一个t-1次多项式$f(x)=a_0+a_1x+\cdots+a_{t-1}x^{t-1}$。
- N个参与者$P_1, \cdots, P_n$, P_i的子密钥为$f(i)$。任意t个参与者得到秘密，可使用$\{(i_1, f(i_1)) \mid 1=1, \cdots, t\}$构造方程组

$$\begin{cases} a_0 + a_1(i_1) + \cdots + a_{t-1}(i_1)^{t-1} = f(i_1) \\ \vdots \\ a_0 + a_1(i_t) + \cdots + a_{t-1}(i_t)^{t-1} = f(i_t) \end{cases}$$

Lagrange插值公式

设$\{(x_1, y_1), \cdots, (x_t, y_t)\}$是平面上t个点构成的点集，其中$x_i$ (i=1, …, t)各不相同，那么在平面上存在唯一的t-1次多项式f (x)通过这t个点. 若把秘密s取做f (0)，n个子密钥取做f (x_i) (i=1, …n), 那么利用其中任意t个子密钥可以重构f (x)，从而可以得到秘密s。

Lagrange插值公式构造多项式

$$f(x)=\sum_{j=1}^{t} f(i_j)\prod_{\substack{l=1\\ l\neq j}}^{t}\frac{(x-i_l)}{i_j-i_l}\pmod q$$

$$s=(-1)^{t-1}\sum_{j=1}^{t} f(i_j)\prod_{\substack{l=1\\ l\neq j}}^{t}\frac{i_l}{i_j-i_l}\pmod q$$

Shamir门限方案的分析

如果t-1个参与者想获得s，可构造t-1个方程，有t个未知量。对任一s_0,设f(0)= s_0，这样可以得到第t个方程，得到f(x)。对每个s_0都有唯一的多项式满足，所以已知t-1个子密钥得不到任何s的信息。因此此方案是完善的。

Shamir门限方案的举例

例：t=3, n=5, q=19, s=11。

随机选a_1=2, a_2=7 $f(x)=7x^2+2x+11 \bmod 19$。

计算：f(1)=1, f(2)=5, f(3)=4, f(4)=17, f(5)=6。

已知f(2), f(3), f(5), 重构：

$$f(x)=5\frac{(x-3)(x-5)}{(2-3)(2-5)}+4\frac{(x-2)(x-5)}{(3-2)(3-5)}+6\frac{(x-2)(x-3)}{(5-2)(5-3)}$$

$$=7x^2+2x+11$$

Asmuth-Bloom门限方案(参数选取)

基于中国剩余定理

令q是一个大素数，$m_1, m_2, \cdots, m_n$是n个严格递增的数，且满足下列条件：

(1) $q > S$

秘密密钥

(2) $(m_i, m_j) = 1 \quad (\forall i, j, i \neq j)$

(3) $(q, m_i) = 1 \quad (i = 1, 2, \cdots, n)$

(4) $N = \prod_{i=1}^{t} m_i > q \prod_{j=1}^{t-1} m_{n-j+1}$

q和n个模数两两互素

Asmuth-Bloom门限方案(秘密分割)

(1) 随机选取整数A满足$0 \le A \le \lfloor N/q \rfloor - 1$，并公布$q$和$A$;

(2) $y = S + Aq$,则有$y < q + Aq = (A+1)q \le \lfloor N/q \rfloor \cdot q \le N$;

(3) 计算$y_i \equiv y (\mathrm{mod}\, m_i)$ $(i = 1, 2, \cdots, n)$。（m_i, y_i）即为一个子共享，将其分别传送给n个用户;

集合$\{(m_i, y_i) \mid i = 1, 2, \cdots, n\}$即构成了一个$(t, n)$门限方案。

Asmuth-Bloom门限方案(秘密恢复)

当t个参与者$i_1, i_2, \cdots, i_t$提供出自己的子份额，

由$\{(m_{i_j}, y_{i_j}) \mid i = 1, 2, \cdots, t\}$建立方程组：

$$\begin{cases} y \equiv y_{i_1} \pmod{m_{i_1}} \\ y \equiv y_{i_2} \pmod{m_{i_2}} \\ \cdots \\ y \equiv y_{i_k} \pmod{m_{i_k}} \end{cases}$$

根据中国剩余定理可求得：

$$y \equiv y' \pmod{N'}$$

其中，$N' = \prod_{j=1}^{t} m_{i_j} \geq N$

由$y' - Aq$即得秘密S。

Asmuth-Bloom门限方案(正确性)

因为由t个成员的共享计算得到的模满足条件$y < N \le N'$，所以$y = y'$是唯一的，再由$y' - Aq$即得秘密S。

另一方面，若仅有t-1个参与者提供自己的子份额(m_i, y_i)，则只能求得$y'' \equiv y (\mathrm{mod}\, N'')$，式中$N'' = \prod_{j=1}^{k-1} m_{i_j}$。

由条件(4)得$N'' < N / q$，即$N / N'' > q$。

令$y = y'' + \alpha N''$，其中$0 \le \alpha < \frac{y}{N''} < \frac{N}{N''}$。由于$N / N'' > q, (N'', q) = 1$，当$\alpha$在$[0, q]$之间变化时，$y'' + \alpha N''$都是$y$的可能取值，因此无法确定$y$。

举例 设秘密 $S=4$，要求构建一个（3，5）门限方案。

（1）参数选取

设选取素数 $q=7$，5 个模数分别为 $m_1=17, m_2=19, m_3=23, m_4=29, m_5=31$。

$N=m_1\cdot m_2\cdot m_3=17\cdot 19\cdot 23=7429>q\cdot m_4\cdot m_5=7\cdot 29\cdot 31=6293$。

（2）秘密分割

在 $[0,\left\lfloor\frac{7429}{7}\right\rfloor-1]=[0,1060]$ 之间随机取 $A=117$，求得 $y=s+Aq=4+117\cdot 7=823$。

然后计算：

$$y_1\equiv y \bmod m_1\equiv 823 \bmod 17\equiv 7$$
$$y_2\equiv y \bmod m_2\equiv 823 \bmod 19\equiv 6$$
$$y_3\equiv y \bmod m_3\equiv 823 \bmod 23\equiv 18$$
$$y_4\equiv y \bmod m_4\equiv 823 \bmod 29\equiv 11$$
$$y_5\equiv y \bmod m_5\equiv 823 \bmod 31\equiv 17$$

$\{(17,7),(19,6),(23,18),(29,11),(31,17)\}$ 即构成一个（3，5）门限方案。

（3）秘密恢复

若第 1、3、5 个成员想恢复秘密，则他们提供自己的共享 $\{(17,7),(23,18),(29,11)\}$，

建立方程组：
$$\begin{cases} y\equiv 7(\bmod 17) \\ y\equiv 18(\bmod 23) \\ y\equiv 11(\bmod 29) \end{cases}$$

由此得：

$$M=17\cdot 23\cdot 29=11339$$
$$M_1=23\cdot 29=667 \qquad M_1^{-1}=13$$
$$M_2=17\cdot 29=493 \qquad M_2^{-1}=7$$
$$M_3=17\cdot 23=391 \qquad M_3^{-1}=27$$

由中国剩余定理得：$y\equiv(7\cdot 667\cdot 13+18\cdot 493\cdot 7+391\cdot 11\cdot 27)(\bmod 11339)\equiv 823$。

所以，恢复秘密为 $S=y-Aq=823-117\cdot 7=4$。

主要内容

- 密钥管理的简介
- 密钥的生命周期
- 公钥证书
- 密钥分配
- 密钥协商
- 密钥托管
- 密钥分割

谢 谢 ！