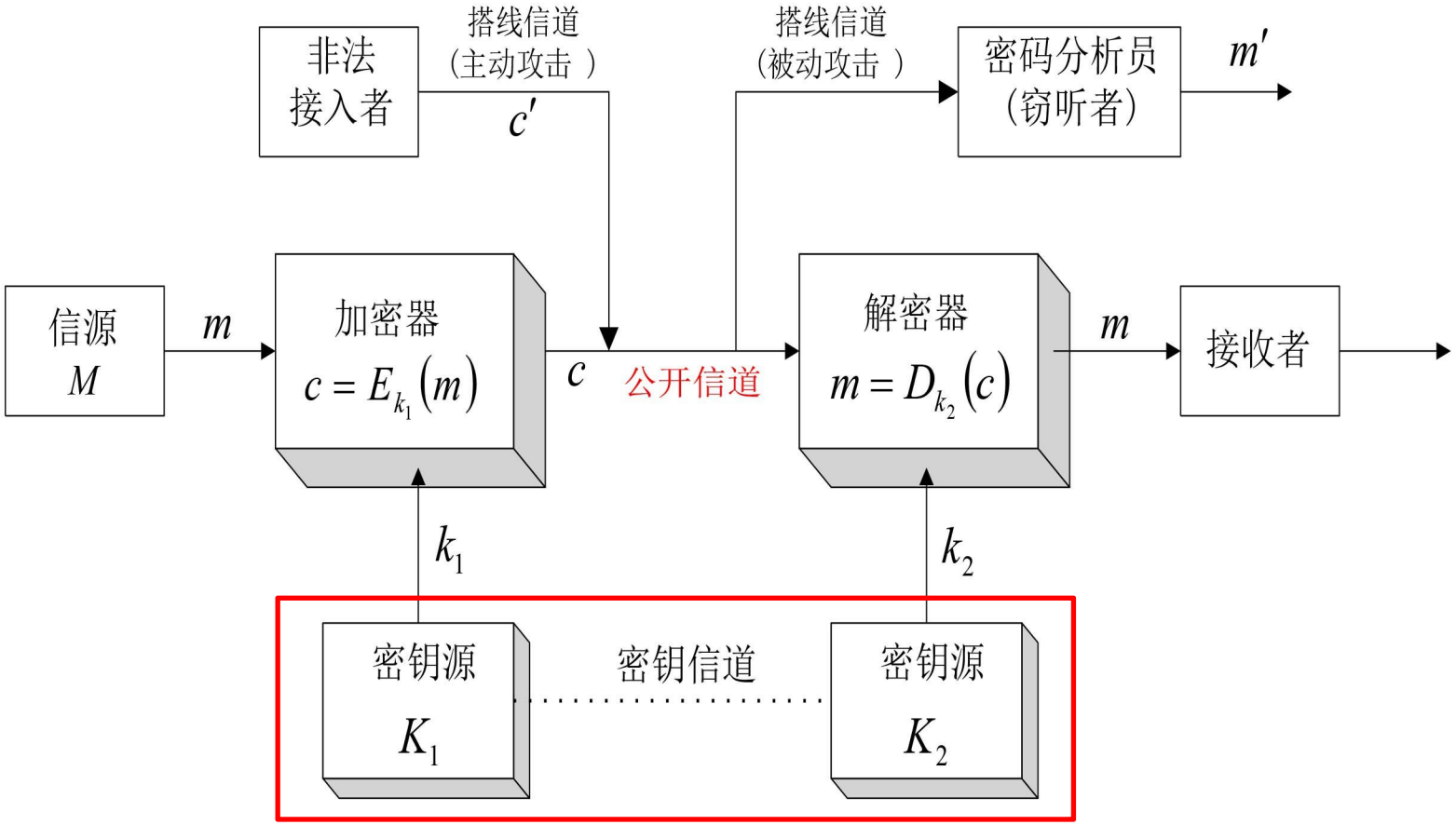
回顾



Kerckhoffs原则：密码算法的安全性完全寓于密钥

第十讲　密钥管理

一、密钥管理概述

二、密钥分配

三、秘密分割

第十讲　密钥管理

一、密钥管理概述

二、密钥分配

三、秘密分割

一、密钥管理概述

- 密钥管理是对密钥从最初产生到最终销毁的全过程进行管理。

- 主要内容

  密钥的产生、分配和维护。其中维护涉及密钥的存储、更新、备份、恢复、销毁等方面。

一、密钥管理概述

（一）密钥管理的层次结构

（二）密钥管理的原则

（三）密钥管理的全过程

1. 密钥的类型

◆根据加密内容的不同

用于**数据加密**的密钥

用于**密钥加密**的密钥

◆根据完成功能的不同

用于验证数字签名的密钥 **公钥**

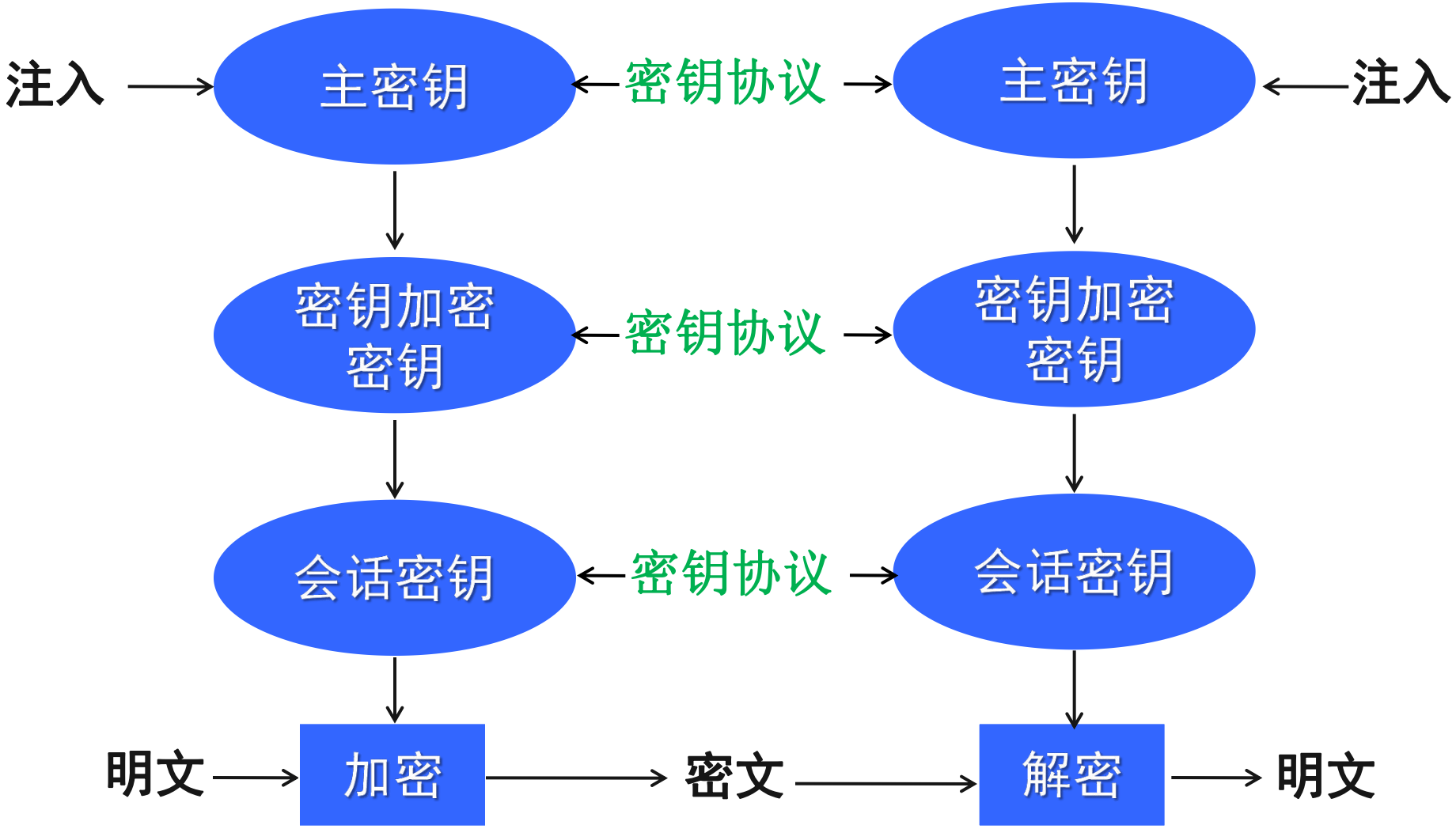
用于实现数字签名的密钥 **私钥**

◆根据密钥的生存周期、功能和保密级别

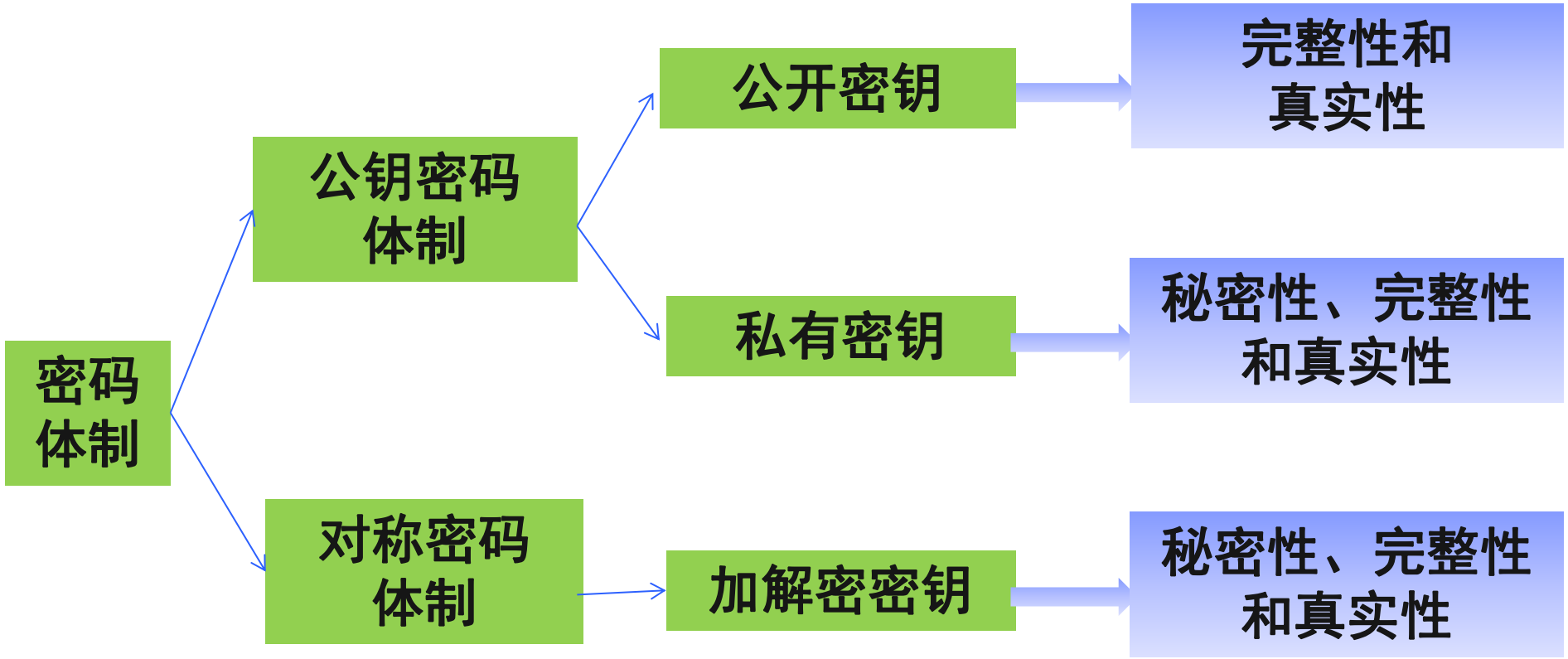
会话密钥、密钥加密密钥和主密钥

一、密钥管理概述

（一）密钥管理的层次结构



2. 密码体制不同，密钥管理也不相同



第十讲　密钥管理

一、密钥管理概述

二、密钥分配

三、秘密分割

二、密钥分配

（一）公钥密码的密钥分配

（二）对称密码的密钥分配

二、密钥分配

公钥密码的密钥分配：分配公钥时，不需要确保其秘密性，但必须确保公钥的**真实性**和**完整性**，绝对不允许攻击者替换或篡改用户的公钥。

分配方法：

- 公开发布
- 公钥管理机构
- 公钥证书

1. 公开发布

- 指用户将自己的公钥发给其他每个用户，或向某一团体广播。

2. 公钥管理机构

- 由一个公钥管理机构对公钥的分配进行严密的控制。公钥管理机构为各用户建立、维护和控制动态的公钥目录

2. 公钥管理机构

公钥管理机构有可能成为系统的瓶颈
公钥目录表容易受到敌手的攻击

3. 公钥证书

- **概念**：公钥证书是一种包含持证主体标识、持证主体公钥等信息，并由可信任的签证机构（CA）签署的信息集合。
- **作用**：证明证书中列出的用户合法拥有证书中列出的公钥。

4. 公钥证书

3. 公钥证书

◆特点

① 用户通过公钥证书交换各自公钥，无须与公钥管理机构联系。

② 公钥证书能以明文的形式进行存储和分配。

③ 公钥证书主要用于确保公钥及其与用户绑定关系的安全。

④ 公钥证书的持证主体可以是人、设备、组织机构或其他主体。

3. 公钥证书

- 产生

用户A　　CA

3. 公钥证书

◆优点

① 由于公钥证书不需要保密，可以在Internet上分发，从而实现公钥的安全分配。

② 公钥证书有CA的签名，攻击者不能伪造合法的公钥证书。

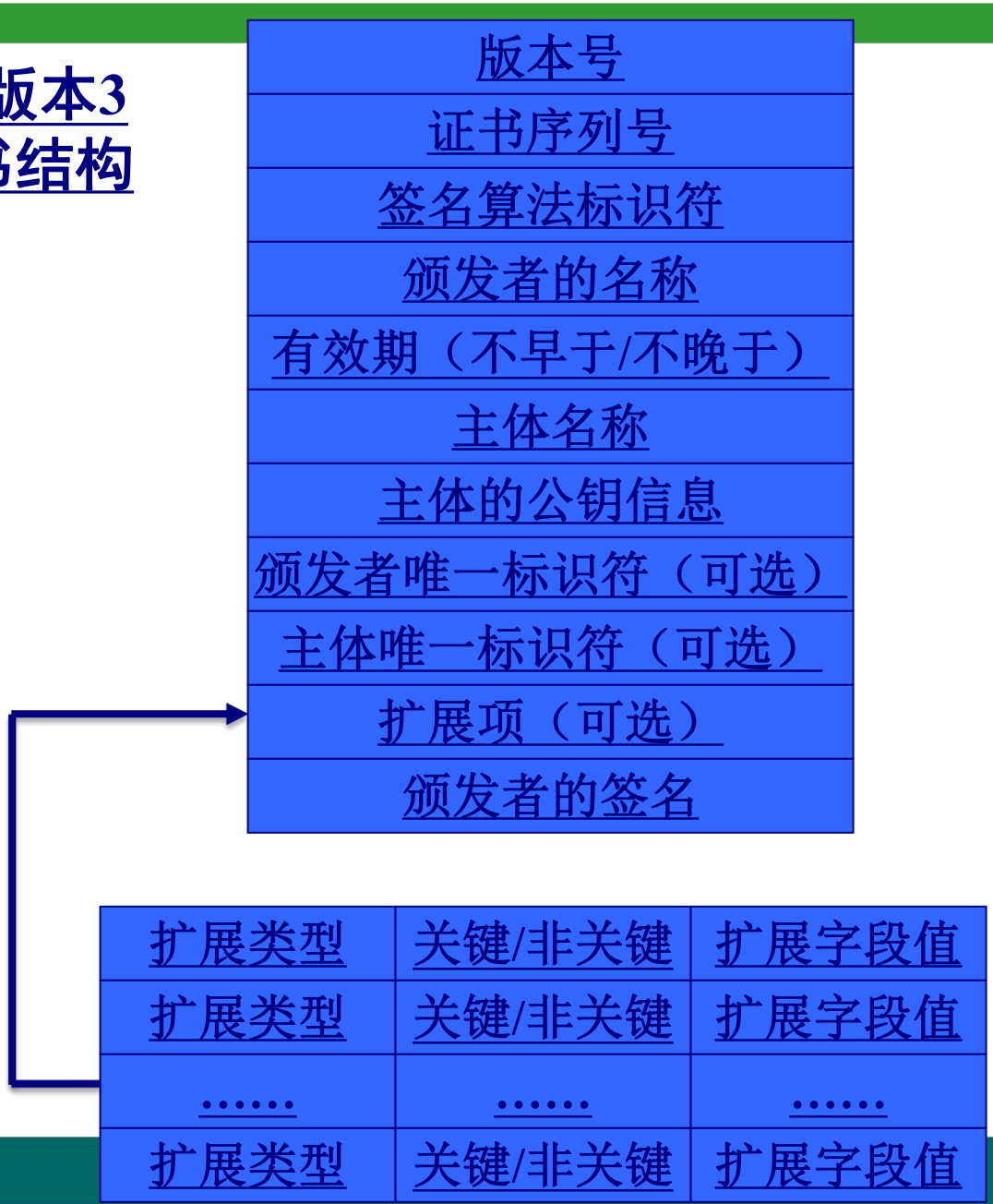
③ 用户只要获得CA的公钥，就可以安全地获得其他用户的公钥

4. X.509证书

- 目前应用最广泛的证书格式是国际电信联盟ITU（International Telecommunication Union）提出的X.509版本3格式。
- X.509标准最早于1988年颁布。在此之后又于1993年和1995年进行过两次修改。
- INTERNET工程任务组（IETF）针对X.509在INTERNET环境的应用，颁布了一个作为X.509子集的RFC2459。从而使X.509在INTERNET环境中得到广泛应用。

二、密钥分配

X.509版本3的证书结构



二、密钥分配

（一）公钥密码的密钥分配

二、密钥分配

（一）公钥密码的密钥分配

二、密钥分配

（一）公钥密码的密钥分配

5. 公钥基础设施

- 公钥证书、证书管理机构、证书管理系统、围绕证书服务的各种软硬件设备以及相应的法律基础共同组成公钥基础设施PKI（Public Key Infrastructure）。
- 公钥基础设施提供一系列支持公钥密码应用（加密与解密、签名与验证签名）的基础服务。
- 本质上，PKI是一种标准的公钥密码的密钥管理平台。

5. 公钥基础设施

- 公钥证书是PKI中最基础的组成部分。

- 此外，PKI还包括签发证书的机构（CA），注册登记证书的机构（RA），存储和发布证书的目录，密钥管理，时间戳服务，管理证书的各种软件和硬件设备，证书管理与应用的各种政策和法律，以及证书的使用者。所有这些共同构成了PKI。

5. 公钥基础设施

1、签证机构CA

- 在PKI中，CA负责签发证书、管理和撤销证书。CA严格遵循证书策略机构所制定的策略签发证书。CA是所有注册用户所信赖的权威机构。
- CA在给用户签发证书时要加上自己的签名，以确保证书信息的真实性。为了方便用户对证书的验证，CA也给自己签发证书。这样，整个公钥的分配都通过证书形式进行。

5. 公钥基础设施

1、签证机构CA

- 对于大范围的应用，一个CA是远远不够的，往往需要许多CA。
  - 例如对于某一行业，国家建立一个最高级的CA，称为根CA。
  - 每个省建立一个省CA，每个地市也都可以建立CA，甚至一个企业也可以建立自己的CA。
  - 不同的CA服务于不同的范围，履行不同的职责。

5. 公钥基础设施

2、注册机构RA

- RA（Registration Authority）是专门负责受理用户申请证书的机构。根据分工，RA并不签发证书，而是负责对证书申请人的合法性进行认证，并决定是批准或拒绝证书申请。
- 证书的签发由CA进行。

5. 公钥基础设施

3、证书的签发

- 经过RA的注册批准后，便可向CA申请签发证书。

CA签发证书的过程如下：

- 用户向CA提交RA的注册批准信息及自己的身份等信息（或由RA向CA提交）；
- CA验证所提交信息的正确性和真实性；
- CA为用户产生密钥（或由用户自己产生并提供密钥），并进行备份；
- CA生成证书，并施加签名；
- 将证书的一个副本交给用户，并存档入库。

5. 公钥基础设施

4、证书的认证

证书认证主要包括以下内容：

①验证证书上的CA签名是否正确。

②验证证书内容的真实性和完整性。

③验证证书是否处在有效期内（由证书里的时间参数来限定有效期）。

④验证证书是否被撤销或冻结。

⑤验证证书的使用方式是否与证书策略和使用限制相一致。

5. 公钥基础设施

5、证书的撤销

- 每个证书都有一个有效使用期限，有效使用期限的长短由CA的政策决定。有效使用期限到期的证书应当撤销。
- 证书的公钥所对应的私钥泄露，或证书的持证人死亡，证书的持证人严重违反证书管理的规章制度等情况下也要撤销证书。
- 和证书的签发一样，证书的撤销也是一个复杂的过程。证书的撤销要经过申请、批准、撤销三个过程。

5. 公钥基础设施

6、信任模型

- 对于大范围的PKI（如一个行业或一个地区，甚至一个国家。），一个CA也是不现实的，往往需要许多CA。
- 这些CA之间应当具有某种结构关系，以使不同CA之间的证书认证简单方便。
  - 拿到1个证书，如何知道是不是伪造的？

6. 公钥基础设施

拿到1个证书，如何知道是不是伪造的？

- 用 CA 的公钥来验证上面的数字签名。

CA 的公钥又是哪里来？

- 从 CA 证书得到的

CA 证书又如何验证是不是伪造的？

- CA 证书上面有另一个 CA 的数字签名。

不断地递归

- 到什么时候才停止？

6. 公钥基础设施

- 验证的过程，停止于自签名证书
- 自签名证书的概念:自己给自己签发证书
- 证书验证的终点其实也是信任的起点

5. 公钥基础设施

以一种安全的、带外的方式，确定用户应该信任哪些信任起点。

- ◆这个过程不是由PKI自己来解决的
- ◆可以是行政命令、合约合同的方式要求用户信任
- ◆或者是用户自愿选择的方式

5. 公钥基础设施

- 信任模型有多种
  - 也就有相应的证书认证路径的构造方法
  - 单个根*CA*
    - 单层模型
    - 层次模型
  - 多个根*CA*
    - *CTL*证书信任列表
    - 交叉认证方式
      - 网状模型
      - 桥CA模型

5. 公钥基础设施

5. 公钥基础设施

5. 公钥基础设施

现实情况是：

- 世界上已经有了很多很多的根CA
- 各根CA也有自己的用户证书
- 如何处理？

6. 公钥基础设施

Certificate Trust List（CTL）

◆仅是客户端软件变化而已

◆*CA*没有做任何工作

查找构造认证路径

6. 公钥基础设施

不同的根*CA*之间相互认证、相互签发证书

◆交叉认证*Cross Certification*

◆交叉证书*Cross Certificate*

交叉认证的结果之一，就是产生交叉证书。

6. 公钥基础设施

- 不同用户有自己的根*CA*，相互交叉认证
  - ◆形成了网状

6. 公钥基础设施

类似于层次模型

◆但是，“*A*认证*C*”与“*C*认证*A*”的路径是不一样的。

◆因为各自的信任起点不一样。

6. 公钥基础设施

桥模型

减少交叉证书的个数

认证路径多了一层

◆相比网状

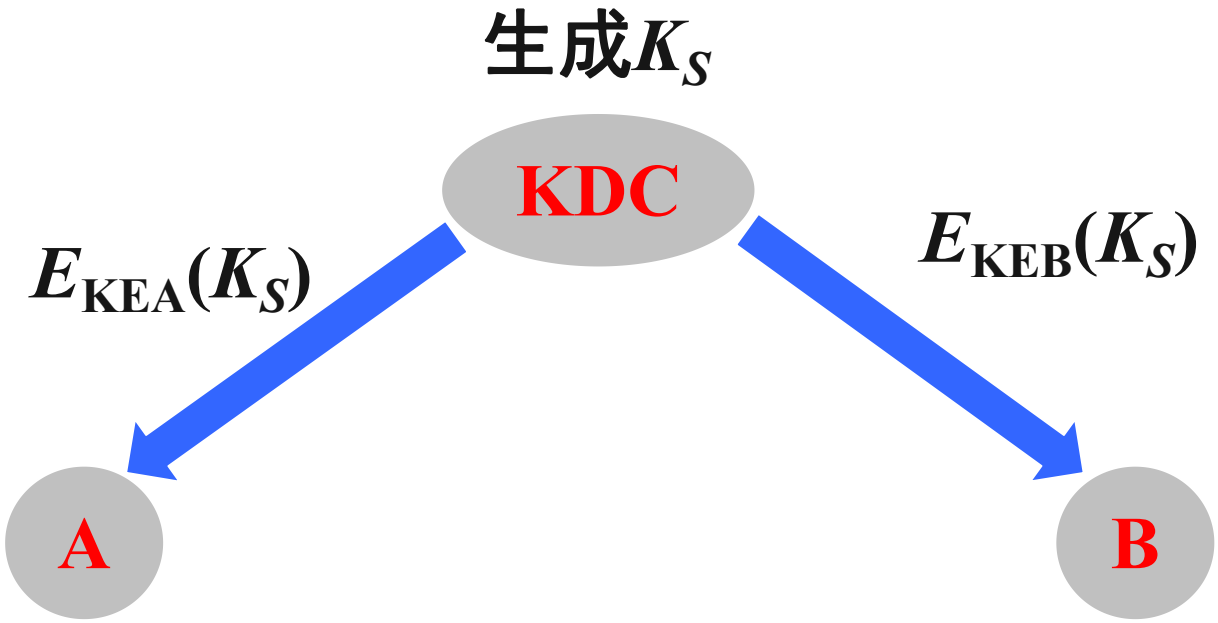
对称密码的密钥分配体系结构大体上可分为

- 密钥分配中心式
- 密钥分配点对点（无中心）式

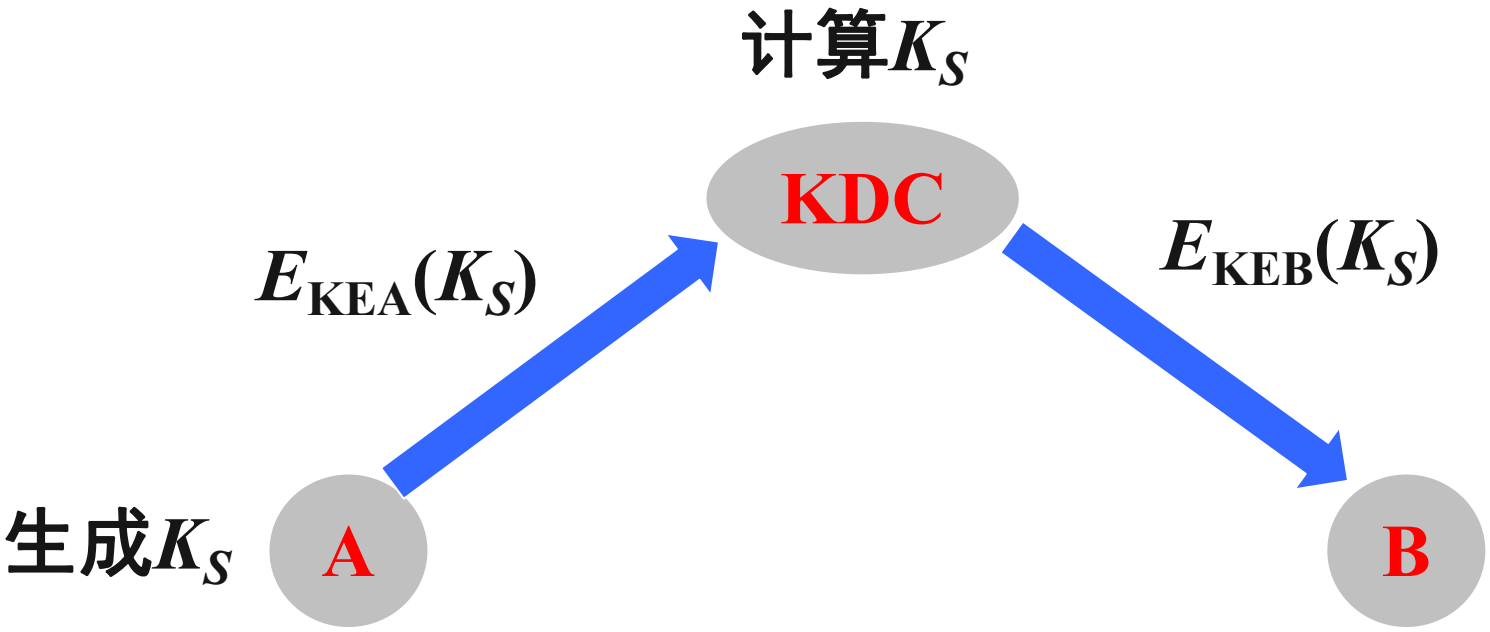
1. 密钥分配中心（KDC）式

- 最基本的密钥分配中心式体系结构是单中心式结构。整个通信网只建立一个密钥分配中心，每个用户U都与KDC共享密钥加密密钥KEU.

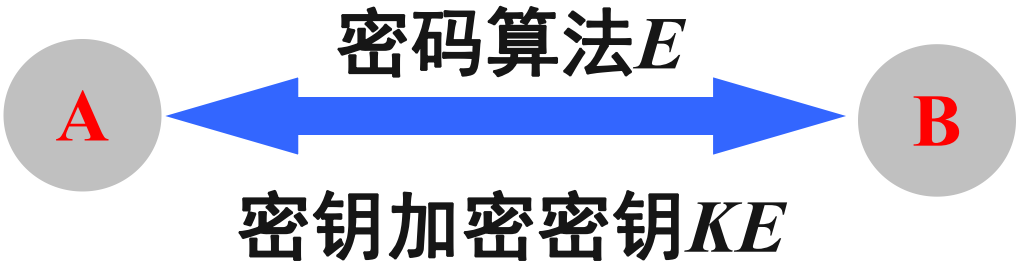
1. 密钥分配中心（KDC）式



1. 密钥分配中心（KDC）式



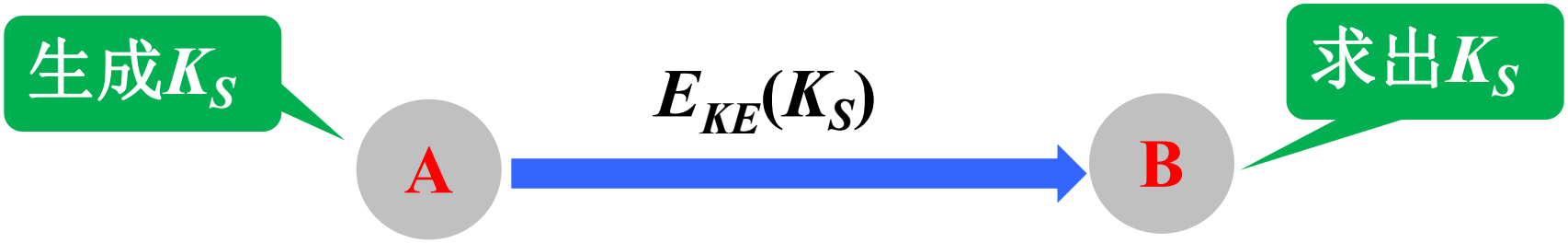
2. 密钥分配点对点（无中心）式



2. 密钥分配点对点（无中心）式

◆当用户A希望与用户B进行保密通信时，可以采用以下三种方式之一来建立共同的会话密钥：

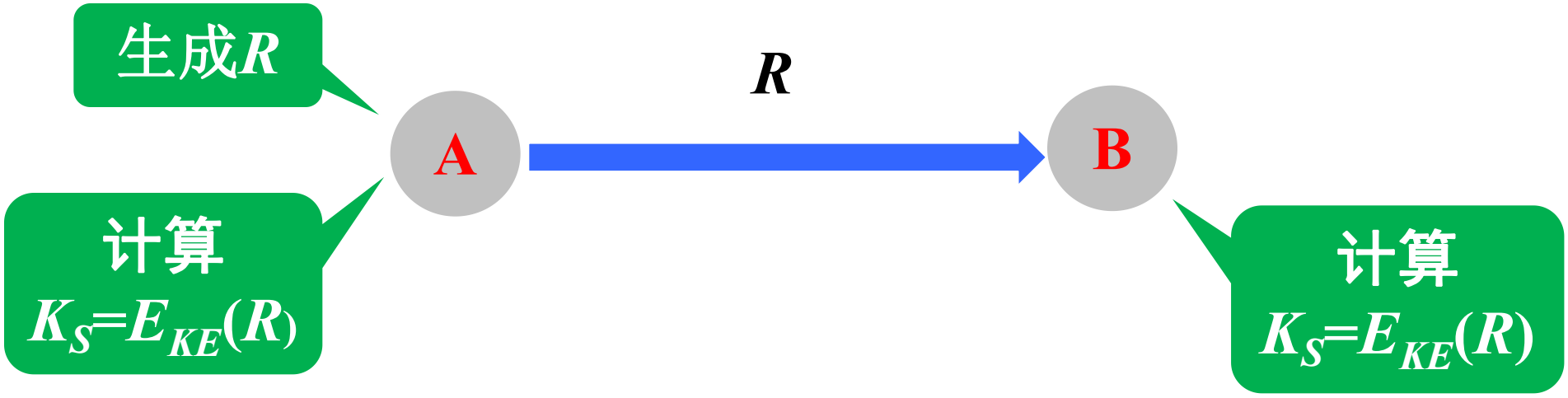
① 密传明用



2. 密钥分配点对点（无中心）式

◆当用户A希望与用户B进行保密通信时，可以采用以下三种方式之一来建立共同的会话密钥：

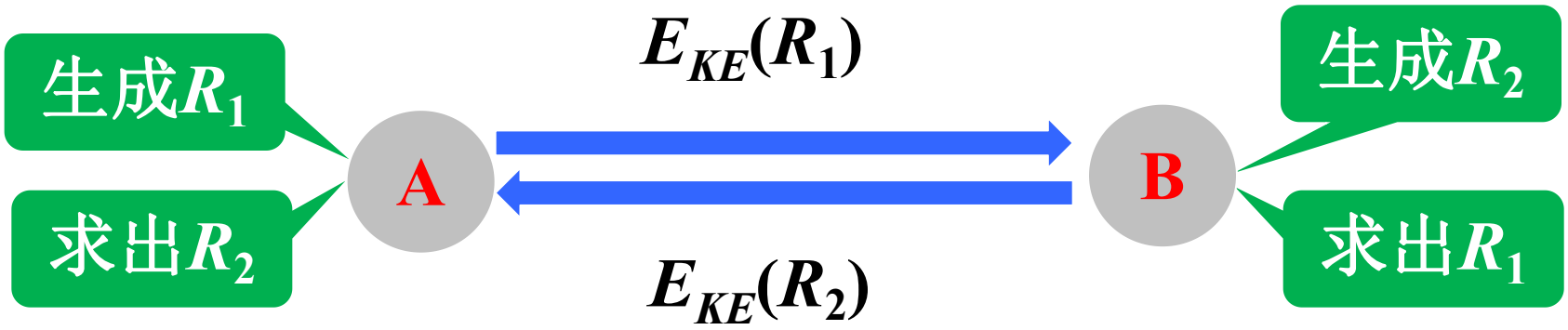
② 明传密用



2. 密钥分配点对点（无中心）式

◆当用户A希望与用户B进行保密通信时，可以采用以下三种方式之一来建立共同的会话密钥：

③ 密钥合成



3. D-H密钥交换协议

- Diffie-Hellman密钥交换是W. Diffie和M. Hellman于1976年提出的第一个公钥密码算法，已在很多商业产品中得以应用。
- 算法目的：

  使得两个用户能够安全地交换密钥，得到一个共享的会话密钥，**算法本身不能用于加解密**。
- 算法的安全性基于离散对数困难问题。

3. D-H密钥交换协议—算法流程

共享的会话密钥：$K= g^{xy} \bmod p$

3. D-H密钥交换协议—实例分析

例1：共享素数p=19，及它的一个本原元g =3

- A选秘密的随机数x=5，B选秘密的随机数y=7
- A计算公开的：$Y_A=g^x=3^5 \bmod 19=15$
- B计算公开的：$Y_B=g^y=3^7 \bmod 19=2$
- A用x计算：$K=Y_B^x=2^5 \bmod 19=13$
- B用y计算：$K=Y_A^y=15^7 \bmod 19=13$
- 共享的会话密钥：$K=13$

课堂练习：在D-H密钥交换协议中，若共享素数p=19，它的一个本原元g =2，试分组完成密钥交换过程。

3. D-H密钥交换协议—算法安全性

◆基于计算离散对数的困难性

∵ 攻击者可利用的信息有

素数p、本原根g、中间值Y_A和Y_B

$$Y_A=g^x \bmod p \quad Y_B=g^y \bmod p$$

$$K= Y_B^x \bmod p = Y_A^y \bmod p = g^{xy} \bmod p$$

若计算K，必须取离散对数先求得x或y

◆除了破获密钥，攻击者还有其他的攻击方式吗？

3. D-H密钥交换协议—算法安全性

第十讲　密钥管理

一、密钥管理概述

二、密钥分配

三、秘密分割

三、秘密分割

例2：某富翁有6个子女，将其遗嘱和存款密码分成6片，每个子女1片。规定至少有4个子女同时出示手中密钥时，就能恢复密码。

例3：某个银行有三位出纳，他们每天都要开启保险库。为防止每位出纳可能出现的监守自盗行为，规定至少有两位出纳在场时才能开启保险库。

三、秘密分割

存贮在系统中的所有密钥的安全性可能最终取决于一个主密钥。这样就存在如下两种安全隐患：

① 若主密钥偶然或蓄意地被暴露，整个系统就易受攻击；

② 若主密钥丢失或毁坏，系统中所有信息就无法使用。

保护主密钥的方法：**秘密分割门限方案**

三、秘密分割

（一）秘密分割门限方案的概念

（二）Shamir门限方案

1. 概念

- 设秘密k被分成n个部分信息，每一部分信息称为一个子密钥或影子，由一个参与者持有，使得
  - ① 由t个或多于t个参与者所持有的部分信息可重构k。
  - ② 由少于t个参与者所持有的部分信息无法重构 k
- 称这种方案为(t, n)-秘密分割门限方案，t称为方案的门限值。

2. 构造方法

- 基于拉格朗日插值多项式的构造：Shamir门限方案
- 基于中国剩余定理的构造：Asmuth-Bloom门限方案
- 基于几何矢量的构造：Blakley门限方案

1. 概念

- Shamir于1979年基于拉格朗日插值多项式提出了一个(t, n)门限方案，在方案中，
  - p是素数
  - k是秘密密钥
- (t, n)门限方案：1个可信中心，n个参与用户。

2. 分配过程

- 可信中心随机选择一个保密的t-1次多项式$h(x)$

$$h(x) = a_0 + a_1 x + \cdots + a_{t-2} x^{t-2} + a_{t-1} x^{t-1} \bmod p$$

$$(a_i \in Z_p, a_0 = k)$$

- 可信中心在Z_p中选择n个非零的、互不相同的元素$x_1, x_2, \ldots, x_n$，计算$y_i = h(x_i)$, $1 \le i \le n$
- 将(x_i, y_i), $1 \le i \le n$分配给n个用户，其中x_i是公开的，y_i是保密的（即子密钥），由n个不同的用户保管。

3. 密钥重构

- 重构密钥就是要确定t-1次多项式，至少需要t个点(x_i, y_i), 少于t个点，就无法确定多项式$h(x)$。
- 子密钥与秘密参数$k,a_1,a_2,\ldots,a_n$的关系为

$$\begin{cases} k + a_1x_1 + a_2x_1^2 + \cdots + a_{t-1}x_1^{t-1} = y_1 \\ k + a_1x_2 + a_2x_2^2 + \cdots + a_{t-1}x_2^{t-1} = y_2 \\ \cdots \quad \cdots \quad \cdots \quad \cdots \quad \cdots \quad \cdots \\ k + a_1x_t + a_2x_t^2 + \cdots + a_{t-1}x_t^{t-1} = y_t \end{cases}$$

3. 密钥重构

◆设方程组的t个未知量为$k, a_1,a_2,\ldots,a_{t-1}$，其矩阵表示为

$$\begin{pmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^{t-1} \\ 1 & x_2 & x_2^2 & \cdots & x_2^{t-1} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 1 & x_t & x_t^2 & \cdots & x_t^{t-1} \end{pmatrix} \begin{pmatrix} k \\ a_1 \\ \cdots \\ a_{t-1} \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ \cdots \\ y_t \end{pmatrix}$$

◆系数矩阵A的行列式$|A|$是一个范德蒙德行列式

$$|A| = \prod_{1 \le j < m \le t} (x_m - x_j) \bmod p \neq 0$$

3. 密钥重构

◆从而可解出未知量$k, a_1,a_2,\ldots,a_{t-1}$为：

$$\begin{pmatrix} k \\ a_1 \\ \cdots \\ a_{t-1} \end{pmatrix} = \begin{pmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^{t-1} \\ 1 & x_2 & x_2^2 & \cdots & x_2^{t-1} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 1 & x_t & x_t^2 & \cdots & x_t^{t-1} \end{pmatrix}^{-1} \begin{pmatrix} y_1 \\ y_2 \\ \cdots \\ y_t \end{pmatrix}$$

◆由于$x_1, x_2, \ldots, x_t$已知，故存在Z_p中已知且不为0的$b_1,b_2,\ldots,b_t$，使得 $k = b_1y_1 + b_2y_2 + \cdots + b_ty_t$

3. 密钥重构

例4：设有5个参与者，密钥k=11，密钥分割者选取素数p=19，Z_p中多项式为：

$$h(x) = 11 + 2x + 7x^2 \bmod 19$$

令x=1, 2, 3, 4, 5 ，计算得到：y=1, 5, 4, 17, 6,可得5个密钥对： (1,1), (2,5), (3,4), (4,17), (5,6)

3. 密钥重构

假设持有子密钥(1,1), (4,17),(5,6)的3个参与者恢复密钥，则需解线性方程组：

$$\begin{cases} k + a_1 + a_2 = 1 \\ k + 4a_1 + 4^2 a_2 = 17 \\ k + 5a_1 + 5^2 a_2 = 6 \end{cases}$$

通过计算可得密钥k=11

3. 密钥重构

- 求多项式的拉格朗日插值公式方法

$$h(x) = a_0 + a_1 x + \cdots + a_{t-2} x^{t-2} + a_{t-1} x^{t-1}$$

$k=a_0=h(0)$

- 利用拉格朗日多项式，$h(x)$可表示为：

$$h(x) = \sum_{s=1}^{t} y_s \prod_{\substack{j=1 \\ j \neq s}}^{t} \frac{x - x_j}{x_s - x_j}$$

3. 密钥重构

◆求多项式的拉格朗日插值公式方法

$$k = h(0) = \sum_{s=1}^{t} y_s \prod_{\substack{j=1 \\ j \neq s}}^{t} \frac{-x_j}{x_s - x_j}$$

令

$$b_s = \prod_{\substack{j=1 \\ j \neq s}}^{t} \frac{-x_j}{x_s - x_j}$$

则

$$k = h(0) = \sum_{s=1}^{t} b_s y_s$$

3. 密钥重构

◆例4：设有5个参与者，密钥k=11，密钥分割者选取素数p=19，Z_p中多项式为：

$$h(x) = 11 + 2x + 7x^2 \bmod 19$$

5个子密钥为： (1,1), (2,5), (3,4), (4,17), (5,6)

$$k = h(0) = \sum_{s=1}^{t} b_s y_s \qquad b_s = \prod_{\substack{j=1 \\ j \neq s}}^{t} \frac{-x_j}{x_s - x_j}$$

3. 密钥重构

$$b_s=\prod_{\substack{j=1\\ j\neq s}}^{t}\frac{-x_j}{x_s-x_j}$$

$$b_1=\left(\frac{-4}{1-4}\right)\left(\frac{-5}{1-5}\right)\bmod 19=5\times 3^{-1}\bmod 19=8$$

$$b_2=\left(\frac{-1}{4-1}\right)\times\left(\frac{-5}{4-5}\right)\bmod 19=(-5)\times 3^{-1}\bmod 19=11$$

$$b_3=\left(\frac{-1}{5-1}\right)\times\left(\frac{-4}{5-4}\right)\bmod 19=1\bmod 19=1$$

(1, 1)
(4, 17)
(5, 6)

$$k=h(0)=\sum_{s=1}^{t}b_s y_s=(8\times 1+11\times 17+1\times 6)\bmod 19=11$$

4. 方案的优缺点

◆ 优点

可以在不改变原来子密钥的前提下，增加新的参与者。

方法：随机选一个x_{n+1}，并将新的子密钥$y_{n+1}=h(x_{n+1})$分配给第$n+1$个参与者。

4. 方案的优缺点

◆ 缺点

对门限方案的攻击，不仅要防敌对的第三方，也要防内部不诚实的参与者。

在Shamir门限方案中，当x_j的值公开时，实际上b_s是已知的，密钥恢复仅与各子密钥有关。

$$b_s = \prod_{\substack{j=1 \\ j \neq s}}^{t} \frac{-x_j}{x_s - x_j} \qquad k = h(0) = \sum_{s=1}^{t} b_s y_s$$

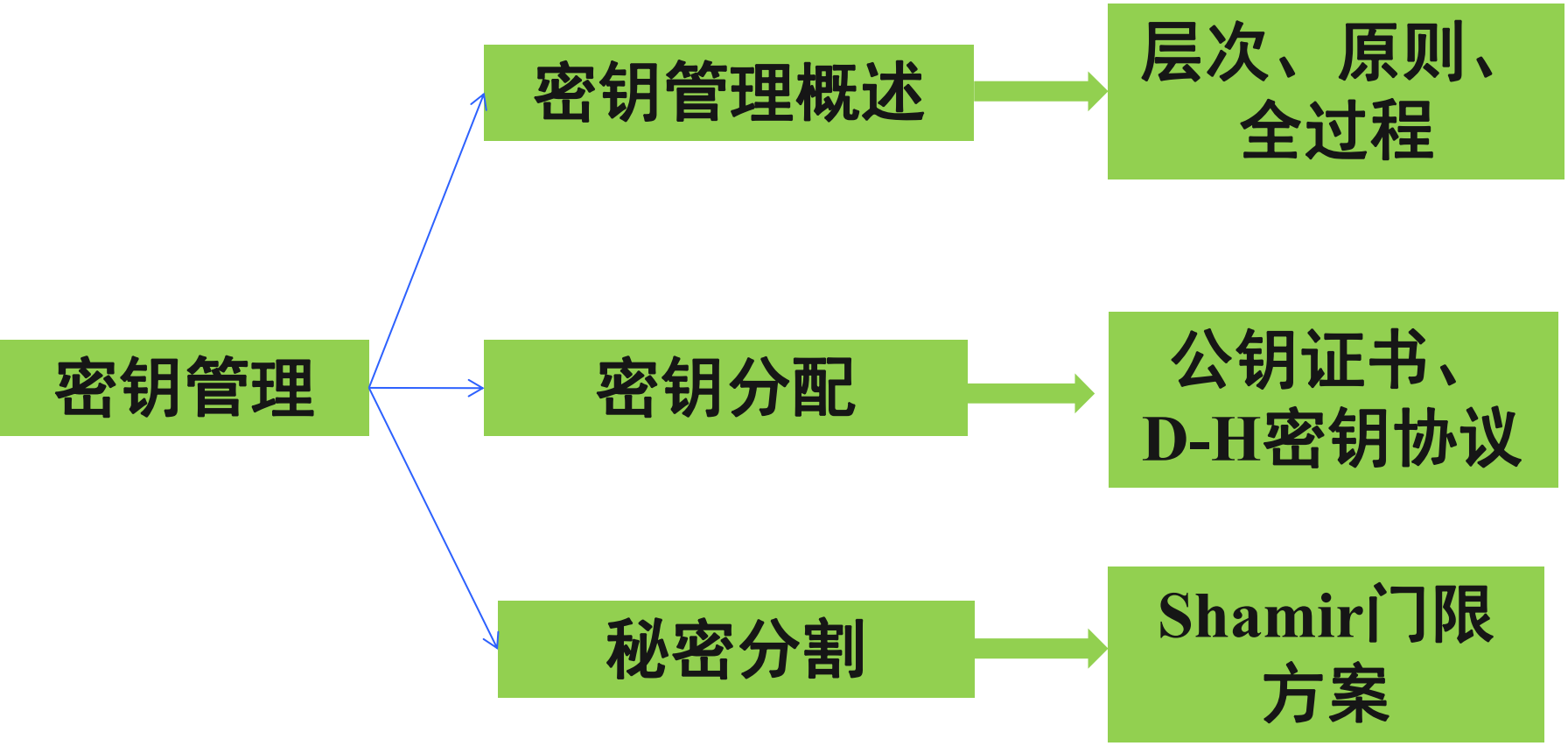
4. 方案的优缺点

◆ 缺点

若不诚实的参与者出示假的子密钥y_j+e，这时恢复出的密钥为$k+eb_j$，由于其他用户无法知道不诚实者所附加的e，因而得不到真正的密钥k的任何消息，但他自己却因知道e，故可恢复出秘密密钥k。

这一缺陷可通过对子密钥的认证来解决。

小结



思考题

1. 设有7个参与者，密钥分割者选取素数p=17，Z_p中多项式$h(x)$为一个4次多项式，并且给出如下7个密钥对：

(1,6), (2,4), (3,11), (4,12), (5,8), (6,16), (7,1)

试用Shamir门限方案恢复密钥k。