

密码学

第九章 密钥管理

网络空间安全学院

朱丹 戚明平

zhudan/mpqi@nwpu.edu.cn

章节安排

Outline



密钥管理概述



传统密码体制的密钥管理



公钥密码体制的密钥管理

章节安排

Outline



密钥管理概述



传统密码体制的密钥管理



公钥密码体制的密钥管理

✦ 密码的公开设计原则：

✦ 密码体制的安全应当只取决于密钥的安全，而不取决于对密码算法的保密

✦ 密钥管理包括密钥的产生、存储、分配、组织、使用、停用、更换、销毁等一系列技术问题

✦ 每个密钥都有其生命周期，要对密钥的整个生命周期的各个阶段进行全面管理





✦ 密码体制不同，密钥的管理方法也不同；密钥管理是一个很困难的问题

✦ 历史表明，从密钥管理环节窃取秘密，要比单纯从破译密码环节窃取秘密所花的代价小得多

✦ 在密码算法确定之后，密钥管理就成为密码应用中最重要的问题



密钥管理的原则

(1) 区分密钥管理的策略和机制

-  **策略是密钥管理系统的高级指导**：策略重在原则指导，而不重在具体实现；策略通常是原则的，简单明了的
-  **机制是实现和执行策略的技术和方法**：机制是具体的、复杂繁琐的
-  没有好的管理策略，再好的机制也不能密钥的安全
-  相反，没有好的机制，再好的策略也没有实际意义

密钥管理的原则

(2) 全程安全原则


-  必须在密钥的产生、存储、分配、组织、使用、停用、更换、销毁的全过程中对密钥采取妥善的安全管理。只有各个阶段都是安全时，密钥才是安全的
-  密钥从一产生到销毁的全过程中，除了在使用的时候可以以明文形式出现外都不应当以明文形式出现

(3) 最小权利原则





-  应当只分配给用户进行某一事物处理所需的最小密钥集合

密钥管理的原则

(4) 责任分离原则

-  一个密钥应当专职一种功能，不要让一个密钥兼几个功能。例如，用于加密的密钥不能用于签名

(5) 密钥分级原则

-  对于一个大的系统，应当采用密钥分级的策略
-  根据密钥的职责和重要性，把密钥划分为几个级别责任分离原则
-  用高级密钥保护低级密钥，最高级的密钥由物理、技术和管理安全保护
-  这样，既可以减少受保护的密钥数量，又可以简化密钥的管理工作

密钥管理的原则

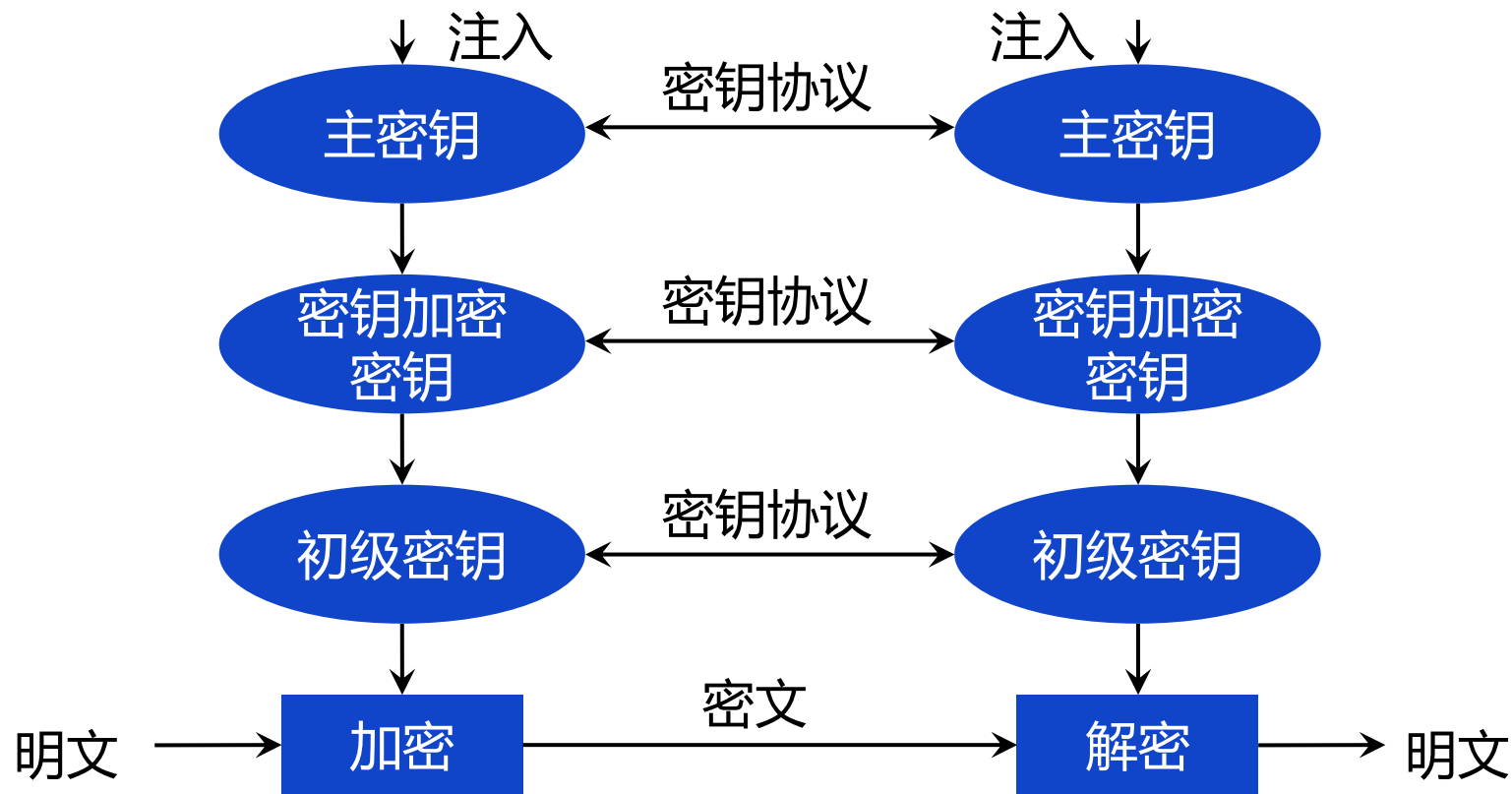
- ✎ (7) 密钥必须满足安全性指标
 - ✎ 密钥应具有足够的长度
 - ✎ 决定密钥长度需综合考虑数据价值、数据安全期需求和攻击者资源情况等因素

- ✎ (8) 密码体制不同，密钥管理也不相同
 - ✎ 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

信息类型	时间	最小密钥长度
战场军事信息	数分钟/小时	56~64 bit
产品发布、合并、利率	几天/几周	64 bit
贸易秘密	几十年	112 bit
氢弹秘密	>40年	128 bit
间谍身份	>50年	128 bit
个人隐私	>50年	128 bit
外交秘密	>65年	至少128 bit

密钥管理的层次结构

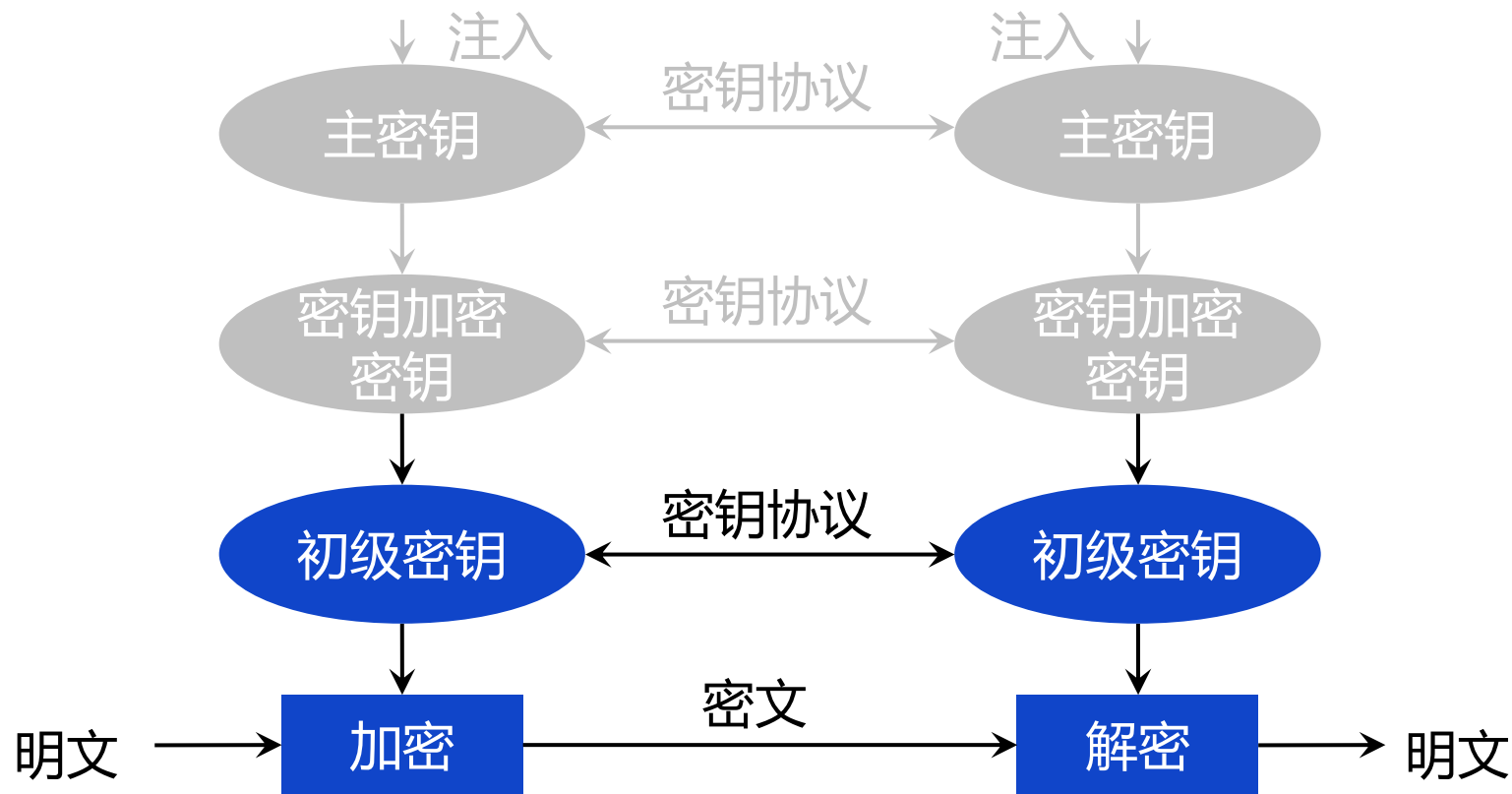
✎ 根据不同种类密钥所起的作用和重要性不同，**现有密码系统的设计大都采用了层次化的密钥结构**，这种层次化结构与对系统的密钥控制关系是对应的。



三层密钥管理的层次结构

密钥管理的层次结构

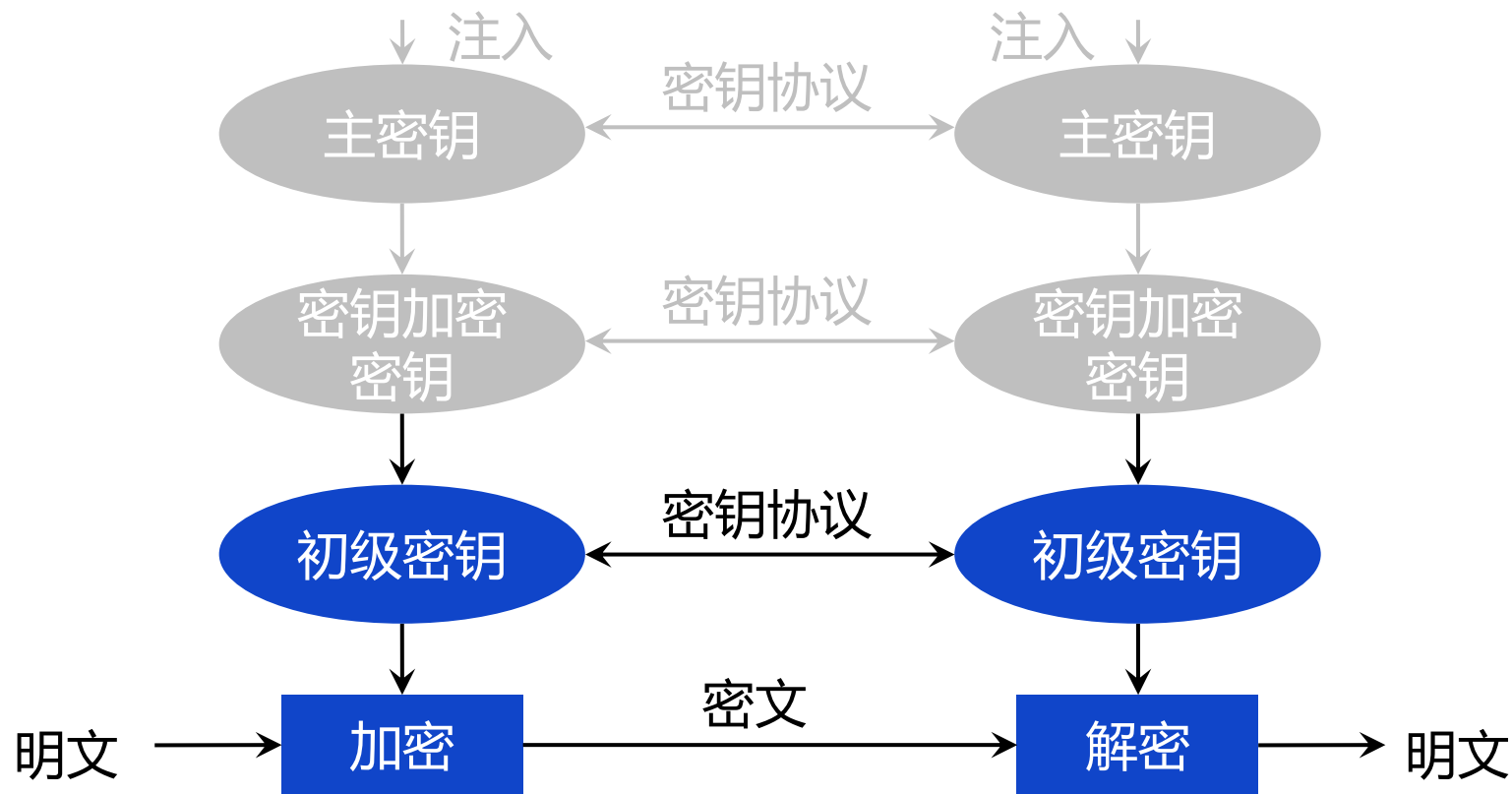
✎ 初级密钥：直接用于加解密数据（通信、文件）的密钥；用于通信保密称为初级通信密钥，用于保护会话称为会话密钥，用于文件保密称为初级文件密钥。可通过硬件或软件方式自动产生，也可用户自己提供。



三层密钥管理的层次结构

密钥管理的层次结构

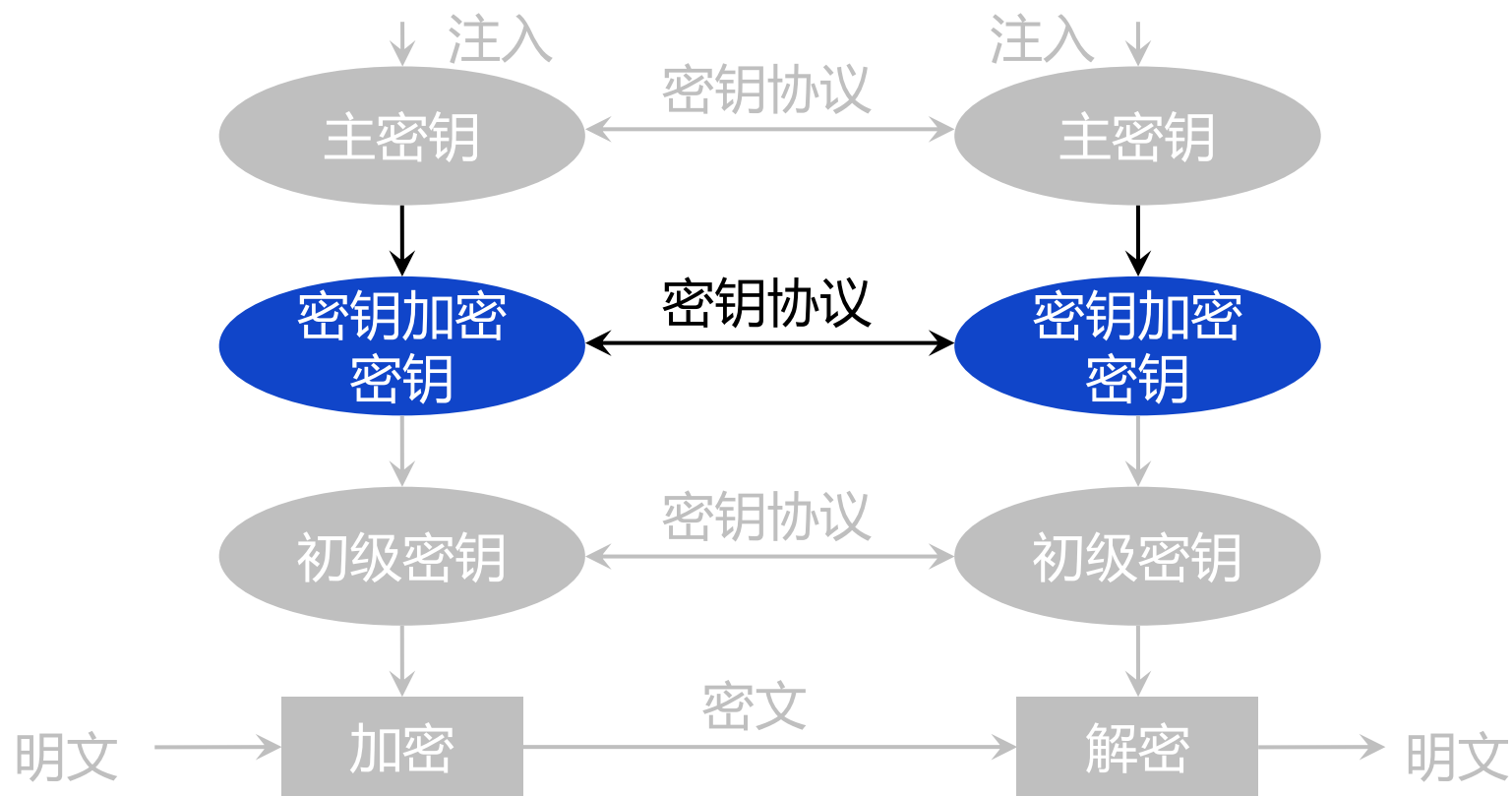
✎ 初级密钥：初级通信/会话密钥原则上采用“一次一密”方式；初级通信密钥的生存周期很短；初级文件密钥与所保护的文件的生存周期一样长；初级密钥必须受更高一级的密钥保护，直到生存周期结束为止。



三层密钥管理的层次结构

密钥管理的层次结构

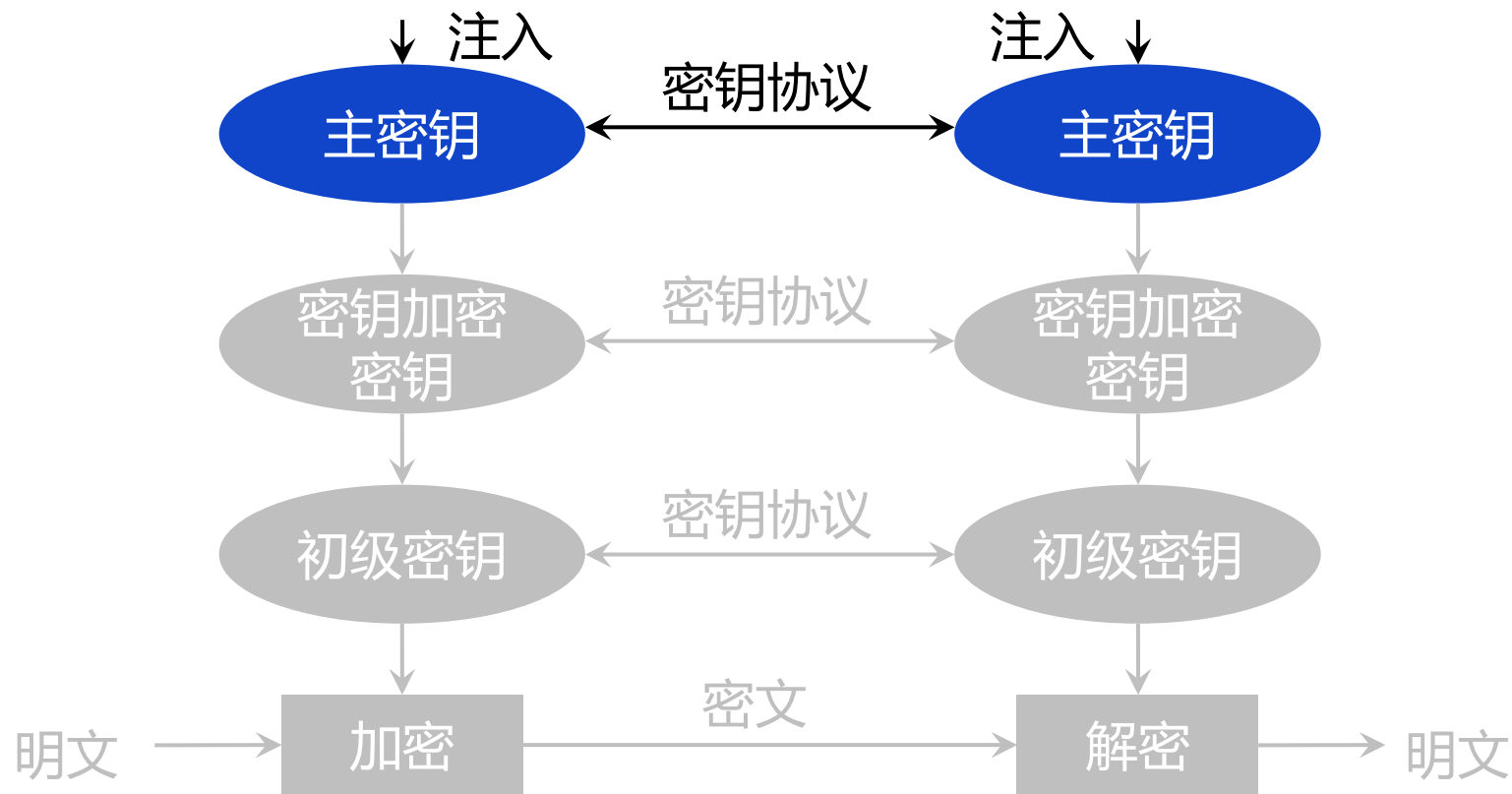
- ✎ 密钥加密密钥：对初级密钥进行加密所使用的密钥，也称次主密钥或二级密钥。可由专职密钥安装人员提供并安装；也可经专职密钥安装人员批准，由系统自动产生。二级密钥的生存周期一般较长；二级密钥必须接受主密钥保护。



三层密钥管理的层次结构

密钥管理的层次结构

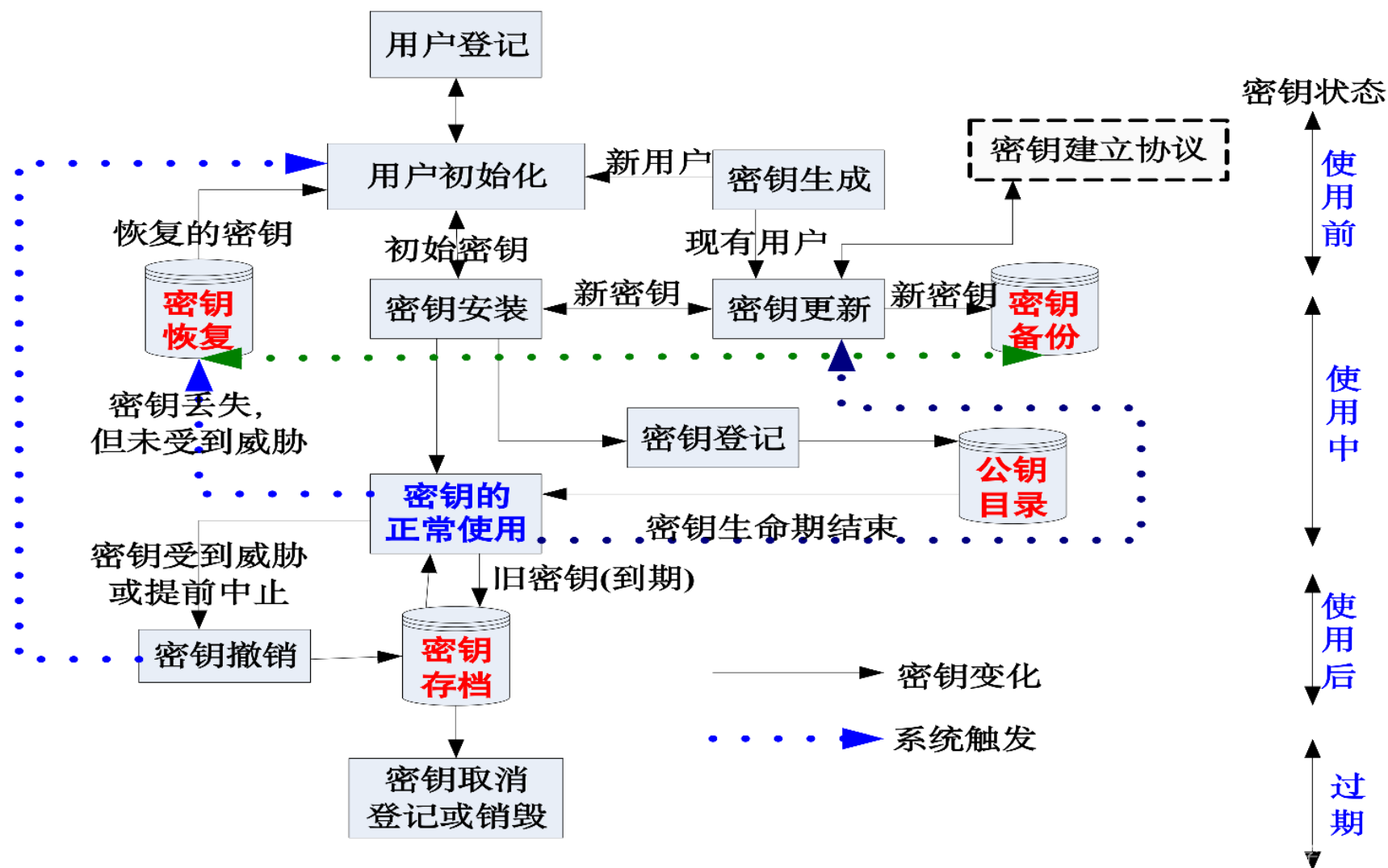
- 主密钥：密钥管理方案中的最高级密钥；用于对密钥加密密钥和初级密钥进行保护；由专职人员产生并妥善安装；生存周期很长；只能以明文形式存储；必须采用安全的物理、技术、管理措施对主密钥进行保护！



三层密钥管理的层次结构

密钥管理全过程

- ❖ 密钥处于4种不同的状态
- ❖ 包含用户登记、用户初始化、**密钥生成**、**安装**、**登记**、**正常使用**、**更新**、**备份**、**恢复**、**存档**、**撤销**、**注销与销毁** 12个重要阶段



章节安排

Outline



密钥管理概述



传统密码体制的密钥管理



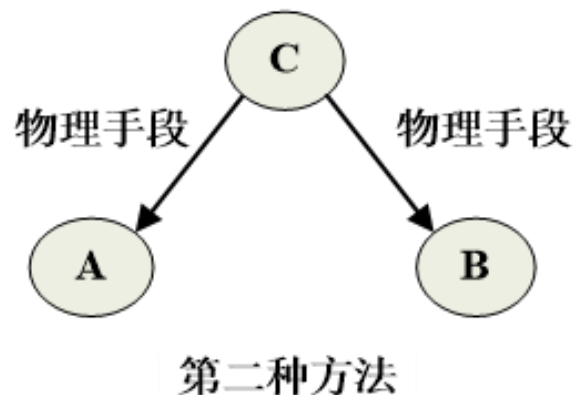
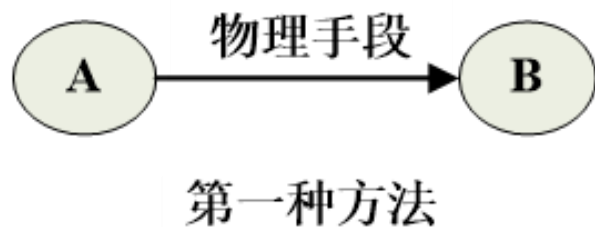
公钥密码体制的密钥管理

密钥分配的基本方法

- ✎ 两个用户（主机、进程、应用程序）在用传统密码体制进行保密通信时，首先必须有一个**共享的秘密密钥**，为防止攻击者得到密钥，还必须**时常更新密钥**。
- ✎ 两个用户A和B获得共享密钥的方法有以下4种：
 - ✎ 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
 - ✎ 如果A和B与**第三方C**分别有一保密信道，则C为A、B选取密钥后，分别在两个保密信道上发送给A、B

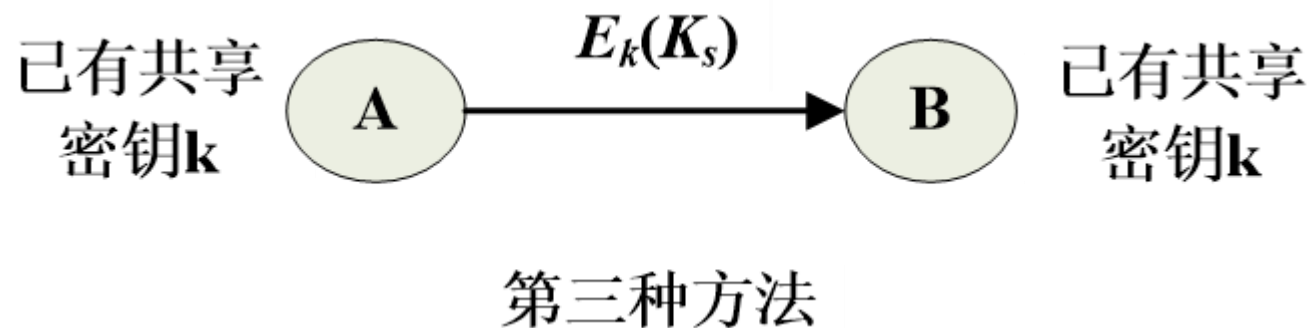
密钥分配的基本方法

- ✎ 第一种和第二种方法称为人工发送
- ✎ 在通信网中，若只有个别用户想进行保密通信，密钥的人工发送还是可行的。然而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，则密钥数目为 $n(n-1)/2$ 。因此当 n 很大时，密钥分配的代价非常大，密钥的人工发送是不可行的
- ✎ 系统的主密钥或初始密钥一般物理手段发送



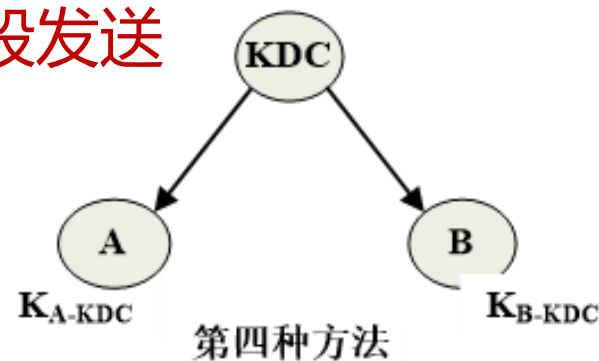
密钥分配的基本方法

- ✎ 对于第三种方法
- ✎ 攻击者一旦获得一个密钥就可获取以后所有的密钥；而且用这种方法对所有用户分配初始密钥时，代价仍然很大



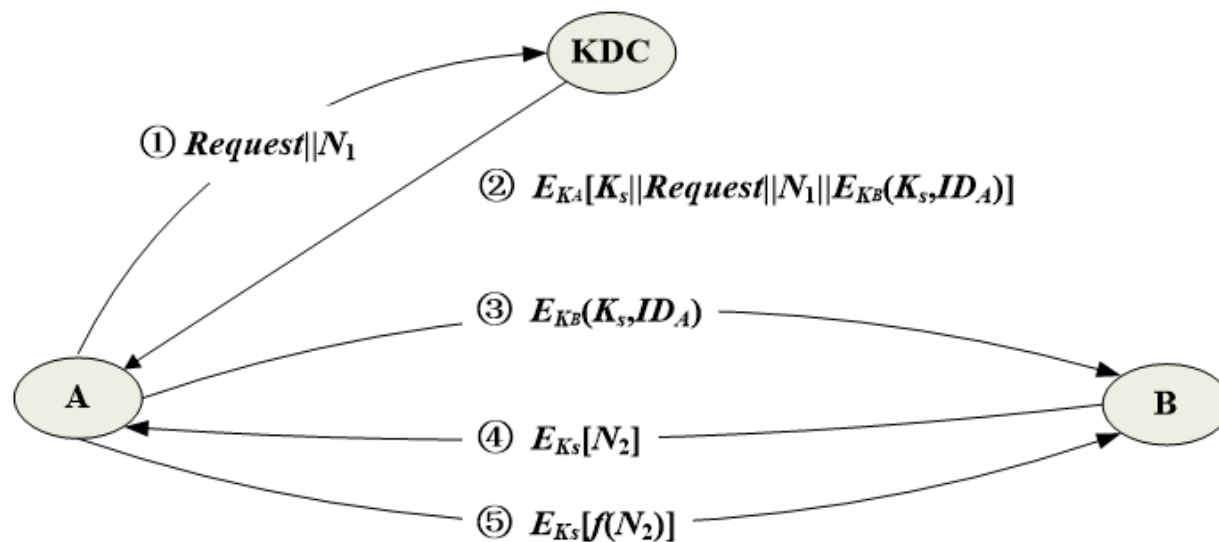
密钥分配的基本方法

- ✦ 第四种方法比较常用
- ✦ 其中的第三方通常是一个负责为用户分配密钥的密钥分配中心 (KDC)
- ✦ 这时每一用户必须和KDC有一个共享密钥, 称为**主密钥** (可通过第二种方法获得)
- ✦ 通过主密钥分配给一对用户的密钥 K_S 称为**会话密钥**, 用于这对用户的保密通信
- ✦ 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, 所以**主密钥可通过物理手段发送**



密钥分配实例

- ① A向KDC发出会话密钥请求。请求由两个数据项组成：A和B的身份；本次业务的唯一标识符 N_1 ；
- ② KDC对A的请求发出应答。应答是由 K_A 加密得到的消息，只有A才能解密读取会话密钥 K_S ；

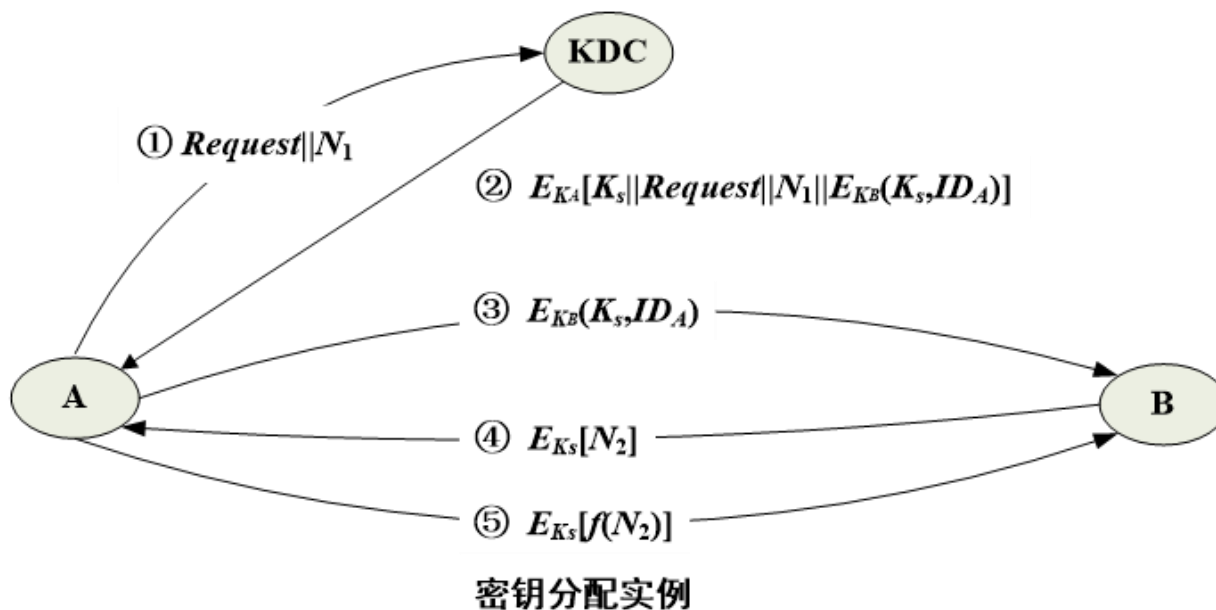


密钥分配实例

密钥分配实例

- ③ A存储会话密钥，并向B转发 $E_{K_B}(K_S, ID_A)$ ，B可以解密得到会话密钥以及A的身份；
- ④ B用一次性会话密钥加密另一个随机数 N_2 ，并将加密结果发送给A；
- ⑤ A以 $f(N_2)$ 的密文作为对B的应答。

步骤④和⑤可使B
相信步骤③收到的
消息不是一个重放



步骤③已完成密钥分
配，步骤④和⑤结合
步骤③执行的是认证
功能

密钥的分层控制

- ✦ 网络中如果用户数目非常多且分布的地域非常广，则需使用多个KDC的分层结构
 - ✦ 在每个小范围（如一个LAN或一个建筑物）内，都建立一个本地KDC；同一范围的用户在进行保密通信时，由本地KDC为他们分配密钥
 - ✦ 如果两个不同范围的用户想获得共享密钥，则可通过各自的本地KDC，而两个本地KDC的沟通又需经过一个全局KDC。这样就建立了两层KDC
 - ✦ 根据网络中用户的数目及分布的地域，可建立3层或多层KDC
- ✦ 分层结构可减小主密钥的分布，因为大多数主密钥是在本地KDC和本地用户之间共享
- ✦ 分层结构还可将虚假KDC的危害限制到一个局部区域，但会降低信任度

会话密钥的有效期

- ✎ 会话密钥更换得越频繁，系统的安全性就越高
 - ✎ 因为敌手即使获得一个会话密钥，也只能获得很少的密文
- ✎ 但是，会话密钥更换得太频繁，将延迟用户之间的交换，同时还造成网络负担
 - ✎ 在决定会话密钥的有效期时，应权衡矛盾的两个方面
- ✎ 对面向连接的协议（如TCP）
 - ✎ 一次会话一密：在连接未建立前或断开时，会话密钥的有效期可以很长。而每次建立连接时，都应使用新的会话密钥
 - ✎ 如果逻辑连接的时间很长，则应定期更换会话密钥

会话密钥的有效期

✎ 面向无连接协议（如用户数据报协议UDP）

✎ 无法明确地决定更换密钥的频率。为安全起见，用户每进行一次交换，都用新的会话密钥。然而这又失去了无连接协议主要的优势，即对每个业务都有最少的费用和最短的延迟。比较好的方案是在某一固定周期内或对一定数目的业务使用同一会话密钥

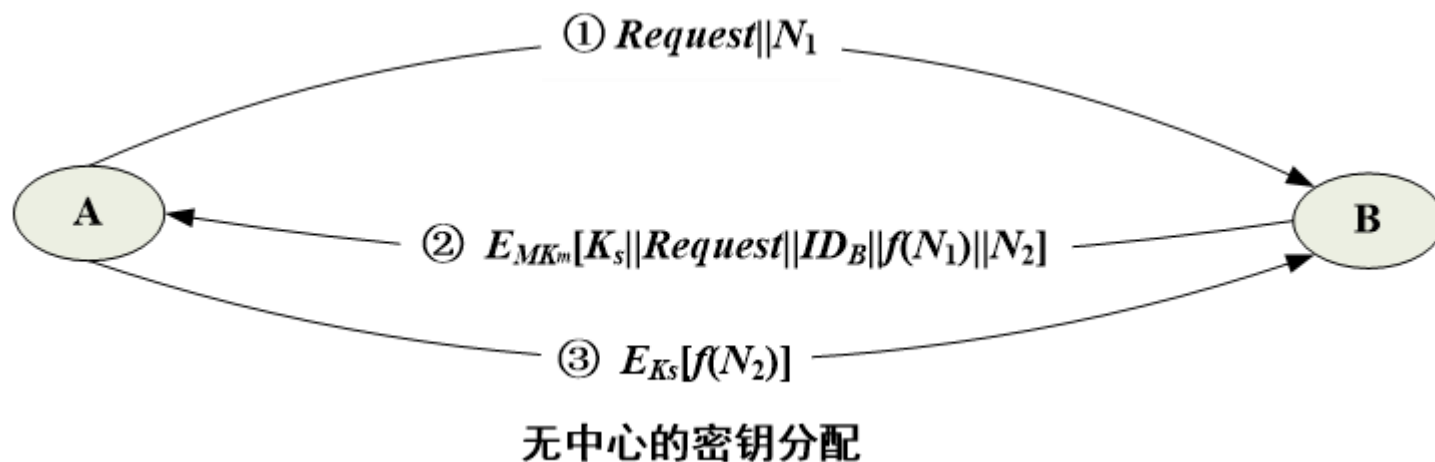
无中心的密钥控制

- ✦ 用密钥分配中心为用户分配密钥时，要求所有用户都信任KDC，同时还要求对KDC加以保护。如果密钥的分配是无中心的，则不必有以上两个要求
- ✦ 然而如果每个用户都能和自己想与之建立联系的另一用户安全地通信，则对有 n 个用户的网络来说，主密钥应多达 $n(n - 1)/2$ 个
- ✦ 当 n 很大时，这种方案无实用价值
- ✦ 但在整个网络的局部范围却非常有用

无中心的密钥控制

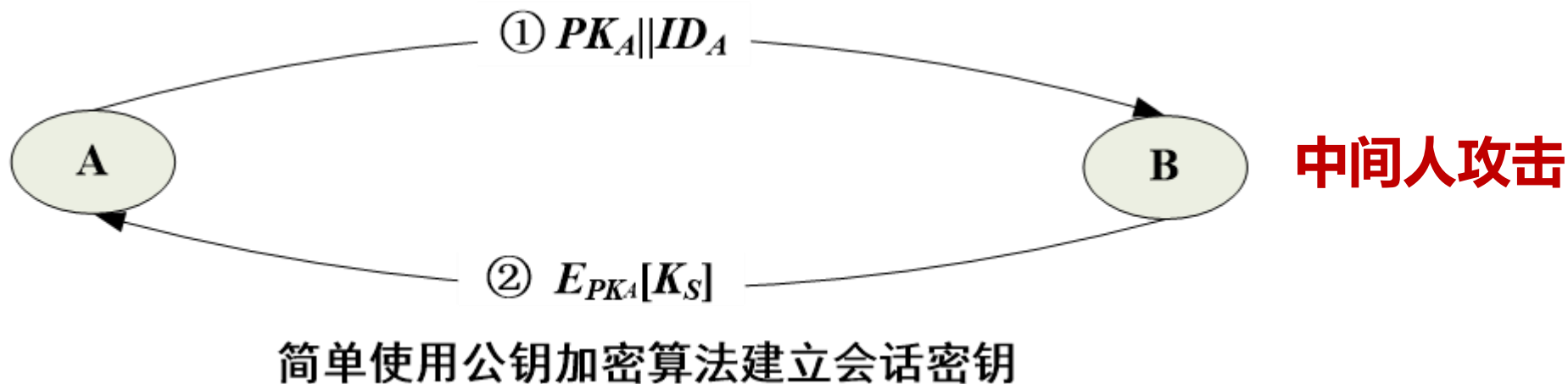
✎ 无中心的密钥分配时，两个用户A和B建立会话密钥需经过以下3步：

- ① A向B发出建立会话密钥的请求和一个一次性随机数 N_1
- ② B用与A共享的主密钥 MK_m 对应答的消息加密，并发送给A
- ③ A使用新建立的会话密钥 K_s 对 $f(N_2)$ 加密后返回给B



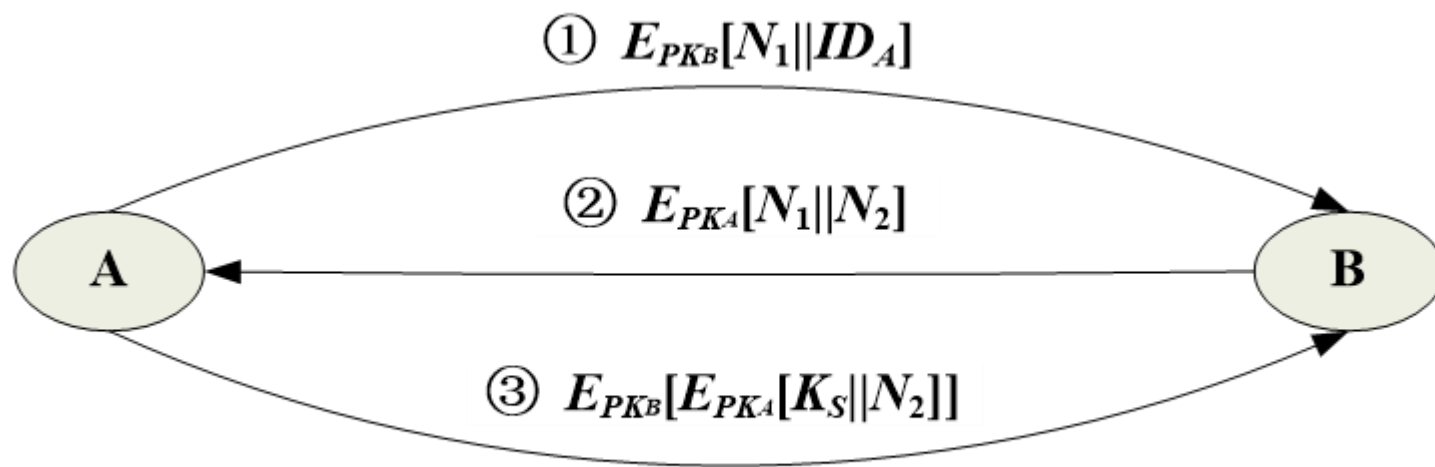
基于公钥密码体制的密钥分配

- 公钥密码学的一个重要优点是便于两个相距遥远的终端用户建立安全的信道，而不需要他们彼此见面或者使用在线认证服务，这正好克服了对称密码技术的缺点
- 使用公钥加密技术，用户A可以直接利用B的公钥对选定的随机会话密钥进行加密并将密文发送给B；B收到密文后即可利用自持的私钥解密得到会话密钥



基于公钥密码体制的密钥分配

- ✎ 加入发送方的身份标识符以及业务的唯一标识符
- ✎ 具有保密性和认证性的密钥分配
- ✎ 既可防止被动攻击，又可防止主动攻击



具有保密性和认证性的密钥分配

章节安排

Outline



密钥管理概述



传统密码体制的密钥管理



公钥密码体制的密钥管理

公钥的分配—公开发布

- ✦ 公开发布指用户将自己的公钥发给每一其他用户，或向某一团体广播
 - ✦ 如PGP (Pretty Good Privacy) 中采用了RSA算法，它的很多用户都是将自己的公钥附加到消息上，然后发送到公开（公共）区域，如因特网邮件列表
- ✦ 缺点很明显，即任何人都可伪造这种公开发布
 - ✦ 如果某个用户假装是用户A并以A的名义向另一用户发送或广播自己的公开钥，则在A发现假冒者以前，这一假冒者可解读所有意欲发向A的加密消息，而且假冒者还能用伪造的密钥获得认证

公钥的分配—公用目录表

公用目录表指一个公用的公钥动态目录表

公用目录表的建立、维护以及公钥的分布由某个可信的实体或组织承担，称这个实体或组织为公用目录的管理员

该方案有以下一些组成部分：

- ① 管理员为每个用户都在目录表中建立一个目录，目录中有两个数据项：用户名和用户的公开钥
- ② 每一用户都亲自或以某种安全的认证通信在管理者那里为自己的公开钥注册，用户能够直接操作目录表

公钥的分配—公用目录表

- ③ 用户如果由于自己的公开钥用过的次数太多或由于与公开钥相关的秘密钥已被泄露，可随时用新密钥替换现有的密钥
- ④ 管理员定期公布或定期更新目录表。例如，像电话号码本一样公布目录表或在发行量很大的报纸上公布目录表的更新
- ⑤ 用户可通过电子手段访问目录表，从管理员到用户必须有安全的认证通信

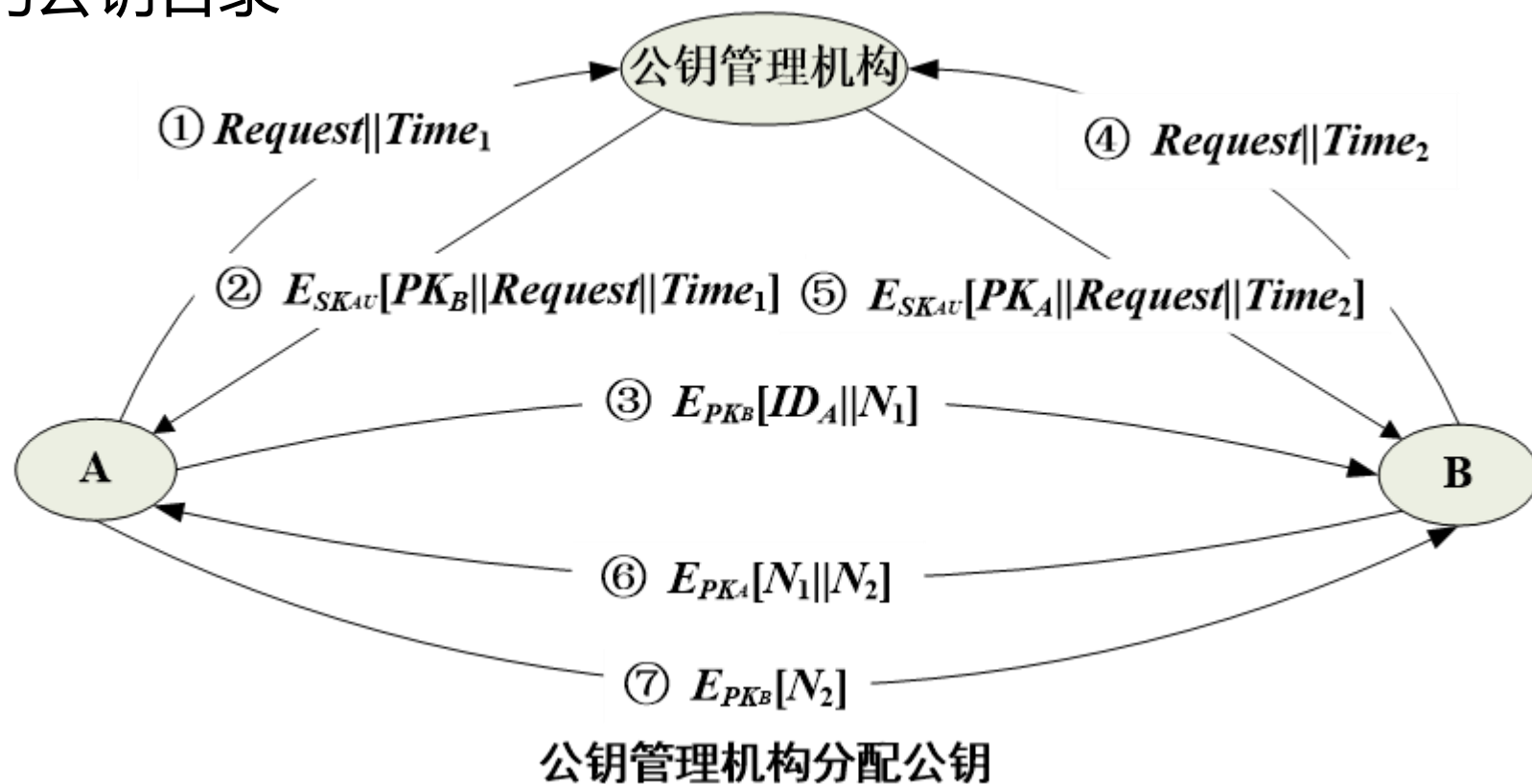
公钥的分配—公用目录表

✎ 安全性高于公开发布，但仍易受攻击

- ✎ 如果敌手成功地获取管理员的秘密钥（密码），就可伪造一个公钥目录表，以后既可假冒任一用户又能监听发往任一用户的消息
- ✎ 公用目录表还易受到敌手的窜扰（破坏）
- ✎ 用户需要登录到公钥目录表中自己查找收方的公钥

公钥的分配—公钥管理机构

- 为防止用户自行对公钥目录表操作所带来的安全威胁，假定有一个公钥管理机构来为各用户建立、维护动态的公钥目录
- 即由用户提出请求，公钥管理机构通过认证信道将用户所需要查找公钥传给用户
- 该认证信道主要基于公钥管理机构的签名



公钥的分配—公钥管理机构

- ① 用户A向管理机构发送一个带时戳的消息，消息包含获取用户B当前公钥的请求
 - ② 管理机构对A的应答由一个消息表示，该消息由管理机构的私钥 SK_{AU} 加密，因此A能用管理机构的公钥解密，并使A相信这个消息的确是来源于管理机构
 - ③ A用 PK_B 对一个消息加密后发往B，包含两个数据项：一是A的身份 ID_A ；二是一次性随机数 N_1 ，用于唯一地标识这次业务
 - ④ 、⑤：B以相同方式从管理机构获取A的公钥
- 用户得到对方的公开钥后保存起来可供以后使用，这样就不必再发送消息①、②、④、⑤了

公钥的分配—公钥管理机构

✎ 用户还可以通过以下两步进行相互认证：

- ⑥ B用 PK_A 加密一次性随机数 N_1 和B产生的一次性随机数 N_2 ，发往A，可使A相信通信的另一方的确是B
- ⑦ A用 PK_B 对 N_2 加密后返回给B，可使B相信通信另一方的确是A

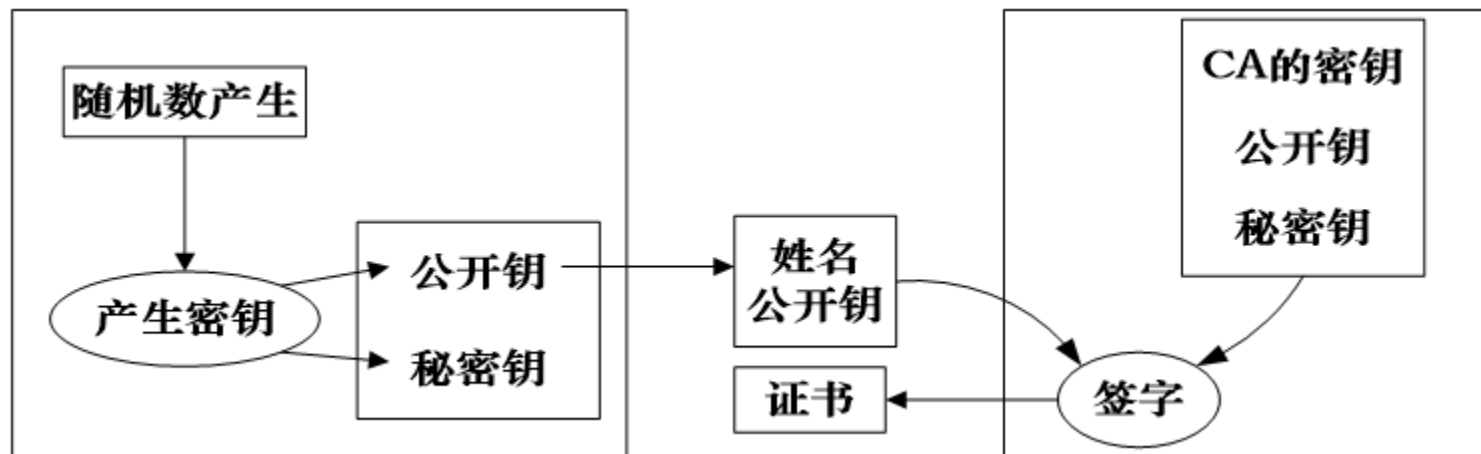
必须定期地通过密钥管理中心获取通信对方的公开钥，以免对方的公开钥更新后无法保证当前的通信

公钥的分配—公钥管理机构

- ✦ 每次密钥的获得由公钥管理机构查询并认证发送，用户不需要查表，提高了安全性
- ✦ 但是，公钥管理机构必须一直在线，由于每一用户要想和他人联系都需求助于管理机构，所以管理机构有可能成为系统的瓶颈
- ✦ 由管理机构维护的公钥目录表也易被敌手通过一定方式窜扰

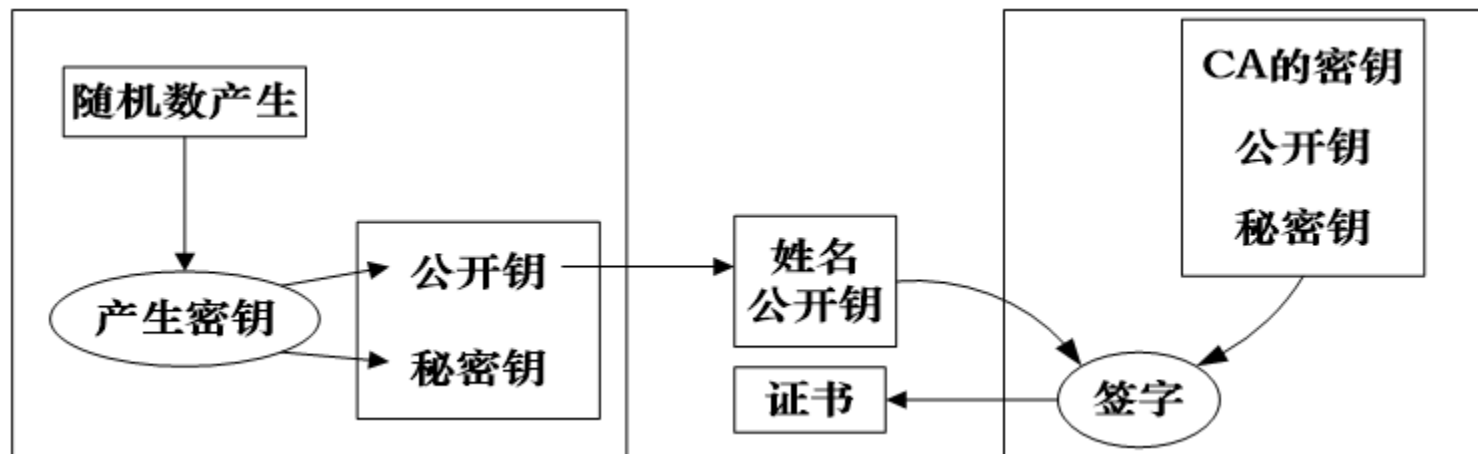
公钥的分配—公钥证书

- ✦ 用户通过公钥证书来互相交换自己的公钥而无须与公钥管理机构联系
- ✦ 公钥证书由证书管理机构CA（Certificate Authority）为用户建立
- ✦ 证书中的数据项有与该用户的密钥相匹配的公钥及用户的身份和时戳等，所有的数据项经CA用自己的密钥签字后就形成证书



公钥的分配—公钥证书

- 证书的形式为 $CA = E_{SK_{CA}}[T, ID_A, PK_A]$
- ID_A 是用户A的身份, PK_A 是A的公钥, T 是当前时戳
- SK_{CA} 是CA的密钥, CA即是为用户A产生的证书



公钥的分配—公钥证书

- ✦ 用户可将自己的公钥通过公钥证书发给另一用户，接收方可用CA的公钥 PK_{CA} 对证书加以验证，即

$$D_{PK_{CA}}[CA] = D_{PK_{CA}} \left[E_{SK_{CA}}[T, ID_A, PK_A] \right] = (T, ID_A, PK_A)$$

- ✦ 因为**只有用CA的公钥才能解读证书**，接收方从而验证了证书的确是由CA发放的，且也获得了发送方的身份 ID_A 和公开钥 PK_A
- ✦ **时戳 T 为接收方保证了收到的证书的新鲜性**，用以**防止发送方或敌方重放**一旧证书。因此**时戳可被当作截止日期**，证书如果过旧，则被吊销

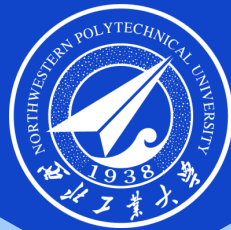
目前应用中典型的密钥管理方案

✎ 基于单钥的密钥分配

- ✎ KDC密钥分发中心，如kerberos认证协议

✎ 基于公钥的密钥分配

- ✎ PKI: CA证书管理机构或公钥管理机构（密钥管理系统开销较大）
- ✎ IDB: 基于身份的公钥体制，用户私钥由KGC计算（存在密钥托管问题）
- ✎ CL-PKC: 无证书的公钥体制，用户私钥由用户和KGC共同计算（公钥由用户产生，并在加密或认证中携带）
- ✎ 自验证的公钥体制（对证书的改进，轻量级证书）



感谢聆听!

THANK YOU FOR YOUR ATTENTION!