(2019春季　课程编号：011184)

信息安全导论

第3章　身份认证

中国科学技术大学 曾凡平

课程回顾：第2章 密码技术

- 2.1 基本概念
- 2.2 对称密码
  - 古典密码
  - 分组密码
  - 序列密码
- 2.3 公钥密码
  - 公钥密码体制原理
  - RSA算法
- 2.4 散列函数和消息认证码
  - 散列函数
  - 消息验证码
- 2.5 数字签名
  - 数字签名简介
  - 基于公钥密码的数字签名原理
  - 数字签名算法
- 2.6 密钥管理
  - 公钥分配
  - 对称密码体制的密钥分配
  - 公钥密码用于对称密码体制的密钥分配
  - Diffie-Hellman密钥交换

第3章 身份认证

身份认证

- Authentication is the act of confirming the truth of an attribute of a single piece of data claimed true by an entity.
- **身份认证：确认某个实体是所声称的实体的行为。**
- 在进行通信之前，必须弄清楚对方是谁，确定对方的身份，以确保资源被合法用户合理地使用。认证是防止主动攻击的重要技术，是安全服务的最基本内容之一。
- 根据被认证实体的不同，身份认证包括两种情况：第一种是**计算机认证人**的身份，称之为**用户认证**；第二种是**计算机认证计算机**，主要出现在通信过程中的认证握手阶段，称之为**认证协议**。

3.1 用户认证

- 用户认证是由计算机对用户身份进行识别的过程，用户向计算机系统出示自己的身份证明，以便计算机系统验证确实是所声称的用户，允许该用户访问系统资源。

- 用户认证的依据主要包括以下三种：
  - (1)所知道的信息，比如身份证号码、账号密码、口令等。
  - (2)所拥有的物品，比如IC卡、USBKey等。
  - (3)所具有的独一无二的身体特征，比如指纹、虹膜、声音等。

3.1.1 基于口令的认证

- 基于用户名 / 口令的身份认证是最简单、最易实现、最易理解和接受的一种认证技术，也是目前应用最广泛的认证方法。
- 包括静态口令和动态口令。

1.静态口令

- 是指**用户口令是静态的**。例如，操作系统及诸如邮件系统等一些应用系统的登录和权限管理，都是采用“用户账户加静态口令”的身份识别方式。口令是一种根据“所知道的信息”实现身份认证的方法，其优势在于实现的简单性，无须任何附加设备，成本低、速度快。

静态口令的认证必须解决两个问题

1)口令存储

- 如果口令以明文方式存储，则易受字典攻击，也就是使用一个预先定义好的单词列表，逐一地尝试所有可能的口令的攻击方式。一般系统的**口令文件存储的是口令的散列值**，即使攻击者得到口令文件，由于散列函数的单向性，也无法得到用户口令。

2)口令传输

- 在网络环境中，基于口令的身份认证系统一般采用客户／服务器模式，如各种Web应用。服务器统一管理多个用户账户，用户口令要从客户机传送到服务器上进行验证。为了保证传输过程中口令的安全，一般**采用双方协商好的加密算法或单向散列函数对口令进行处理后传输**。

静态口令的认证方式存在的安全问题

① 它是一种单因素的认证方式，安全性全部依赖于口令，口令一旦被泄露，用户即可被冒充。

② 为了便于记忆，用户往往选择简单、容易被猜测的口令，如生日。这使得口令被攻击的难度大大降低。

③ 口令在网络上传输的过程中可能被截获。

④ 系统中所有用户的口令以文件形式存储在认证方，攻击者可以利用系统中存在的漏洞获取系统的口令文件。

⑤ 用户在访问多个不同安全级别的系统时，都要求用户提供口令，用户为了记忆的方便，往往采用相同的口令。

⑥ 口令方案无法抵抗重放攻击。

⑦ 只能进行单向认证，即系统可认证用户，而用户无法对系统进行认证，攻击者可能伪装成系统骗取口令。

2.动态口令

- 为了有效地改进口令认证的安全性，人们提出了各种基于动态口令的身份识别方法。
- 动态口令又叫做一次性口令，是指在用户登录系统进行身份认证的过程中，送入计算机系统的**验证数据是动态变化的**。

- 动态口令的主要思路是在登录过程中加入不确定因素，如时间。
- 系统执行某种加密算法**E(用户名+密码+时间)**，**产生一个无法预测的动态口令**，以提高登录过程安全性。

动态口令的产生

1)共享一次性口令表

- 系统和用户共享一个秘密口令表，每个口令只使用一次。用户登录时，系统需要检查用户的口令是否使用过。

2)口令序列

- 用户拥有一个长度为N、单向的、根据某种单向算法前后相关的口令序列，每个口令只使用一次，而计算机系统只用记录一个口令，假设是第M个。用户用第M-1个口令登录时，系统用单向算法计算第M个口令，并与自己保存的第M个口令比对，实现对用户的认证。用户登录N次后，必须重新初始化口令序列。

动态口令的产生

3)挑战—响应方式

- 用户登录时，系统产生一个随机数发送给用户。用户使用某种单向算法将自己的口令和随机数混合起来运算，结果发送给系统。系统用同样的方式进行运算，并通过结果比对实现对用户的认证。

4)时间—事件同步机制

- 这种方式可以看做挑战—响应方式的变形，区别在于以用户登录时间作为随机因素。这种方式要求双方的时间要同步。

基于电子令牌卡生成口令的工作原理

- 用户和计算机系统之间共享同一个用户口令。用户还拥有一种叫做动态令牌的专用硬件，内置电源、密码生成芯片和显示屏，密码生成芯片运行专门的密码算法。
- 当用户向认证系统发出登录请求时，认证系统向用户发送挑战数据。挑战数据通常是由两部分组成的，一部分是种子值，它是分配给用户的在系统内具有唯一性的一个数值，而另一部分是随时间或次数不断变化的数值。
- 用户接收到挑战后，将种子值、随机数值和用户口令输入到动态令牌中进行计算，并把结果作为应答发送给远程认证系统。远程认证系统使用相同的算法和数据进行计算，与从用户那里接收到的应答数据作对比，认证用户的合法性。

动态口令

- 动态口令具有以下几个技术特点：

① 动态性，登录口令是不断变化的。

② 随机性，口令的产生是随机的，具有不可预测性。

③ 一次性，每个口令只使用一次，以后不再使用。

④ 方便性，用户不需记忆口令。

FTGS132

3.1.2 基于智能卡的认证

- 智能卡(smart card)是一种集成的带有智能的电路卡，内置可编程的微处理器，可存储数据，并提供硬件保护措施和加密算法。
- 在智能卡中存储用户个性化的秘密信息，同时在验证服务器中也存放该秘密信息，进行认证时，用户输入PIN(个人身份识别码)，智能卡认证PIN成功后，即可读出智能卡中的秘密信息，进而利用该秘密信息与主机之间进行认证。
- 其中，基于USB Key的身份认证是当前比较流行的智能卡身份认证方式。

USB Key

- USB Key是一种USB接口的硬件设备。它内置单片机或智能卡芯片，有一定的存储空间，可以存储用户的私钥以及数字证书，利用USB Key内置的公钥算法实现对用户身份的认证。由于用户私钥保存在密码锁中，理论上使用任何方式都无法读取，因此保证了用户认证的安全性。
- USB Key结合了现代密码学技术、智能卡技术和USB技术，具有以下4个主要特点。
- **(1)双因子认证。**每一个USB Key都具有硬件PIN码保护，PIN码和硬件构成了用户使用USB Key的两个必要因素，即所谓“双因子认证”。

USB Key的特点

- **(2)带有安全存储空间。**USB Key具有8～128kB的安全数据存储空间，可以存储数字证书、用户密钥等秘密数据，对该存储空间的读写操作必须通过程序实现，用户无法直接读取。其中用户私钥是不可导出的，杜绝了复制用户数字证书或身份信息的可能性。
- **(3)硬件实现加密算法。**USB Key内置CPU或智能卡芯片，可以实现数据摘要、数据加解密和签名的各种算法，加解密运算在USB Key内进行，保证了用户密钥不会出现在计算机内存中，从而杜绝了用户密钥被黑客截取的可能性。
- **(4)便于携带、安全可靠。**如拇指般大的USB Key非常便于随身携带，并且密钥和证书不可导出；USB Key的硬件不可复制，更显安全可靠。

基于USB Key的身份认证主要方式

(1)基于挑战／应答的双因子认证方式

- 先由客户端向服务器发出一个验证请求，服务器接到此请求后生成一个随机数（此为挑战）并通过网络传输给客户端。客户端将收到的随机数通过USB接口提供给计算单元，由计算单元使用该随机数与存储在安全存储空间中的密钥进行运算并得到一个结果（此为应答）作为认证证据传给服务器。
- 与此同时，服务器也使用该随机数与存储在服务器数据库中的该客户密钥进行相同运算，如果服务器的运算结果与客户端回传的响应结果相同，则认为客户端是一个合法用户。
- 密钥运算分别在硬件计算单元和服务器中运行，不出现在客户端内存中，也不在网络上传输，从而保护了密钥的安全，也就保护了用户身份的安全。

基于USB Key的身份认证主要方式

(2)基于数字证书的认证方式

- 随着PKI技术日趋成熟，许多应用中开始使用数字证书进行身份认证与数据加密。
- 数字证书是由权威公正的第三方机构（即CA中心）签发的，以数字证书为核心的加密技术，可以对网络上传输的信息进行加密、解密、数字签名和签名验证，确保网上传递信息的机密性、完整性，以及交易实体身份的真实性，签名信息的不可否认性，从而保障网络应用的安全性。
- USB Key作为数字证书的存储介质，可以保证数字证书不被复制，并可以实现所有数字证书的功能。

USB Key实物

3.1.3 基于生物特征的认证

- 基于生物特征识别的认证方式以人体具有的唯一的、可靠的、终生稳定的生物特征为依据，利用计算机图像处理和模式识别技术来实现身份认证。生物特征识别技术目前主要利用指纹、声音、虹膜、视网膜、脸形、掌纹这几个方面的特征进行识别。
- 与传统的身份认证技术相比，基于生物特征的身份认证技术具有以下优点：
  - ➢不易遗忘或丢失。
  - ➢防伪性能好，不易伪造或被盗。
  - ➢“随身携带”，方便使用。
- 目前，已有的生物特征识别技术主要有指纹识别、掌纹识别、手形识别、人脸识别、虹膜识别、视网膜识别、声音识别和签名识别等。

指纹识别

- 指纹识别是最早研究并利用的，且是最方便、最可靠的生物识别技术之一。
- 指纹识别主要包括三个过程：指纹图像读取、特征提取、比对。

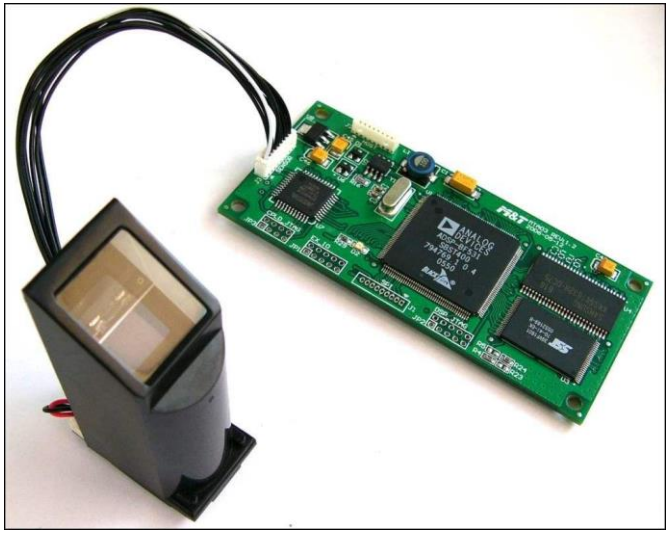
1. 首先，通过指纹读取设备读取到人体指纹的图像，进行初步的处理，使之清晰；
2. 然后，通过指纹图像进行指纹特征数据的提取，这是一种单方向的转换；
3. 最后，计算机通过某种指纹匹配算法进行比对，得到两个指纹的匹配结果。

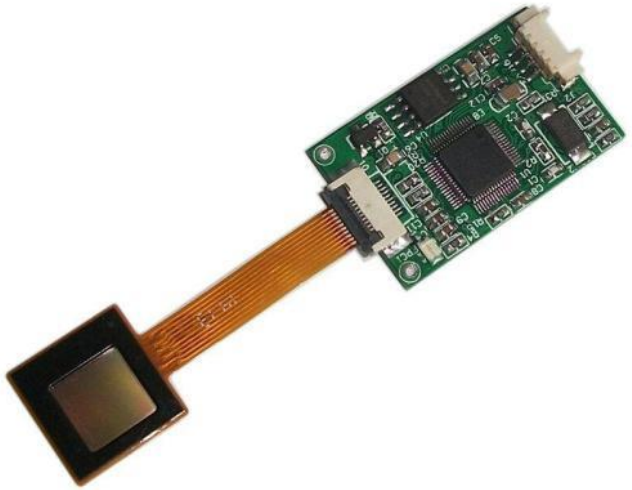
其他的用于认证的生物特征

- 此外，人们还研究了其他的一些生物特征，如手部静脉血管模式、DNA、耳形、身体气味、击键的动态特性、指甲下面的真皮结构等。这些技术与上述的几大技术相比，普遍性较差，主要用于一些特定的应用领域。

- 尽管生物学特征的身份验证机制提供了很高的安全性，但其生物特征信息采集、认证装备的成本较高，只适用于安全级别比较高的场所。

基于生物特征的认证：实例





3.2 认证协议

- 认证协议通过一定的过程，保证使合法的协议一方（或双方彼此）确信对方确实是其所声称的那个实体。
- 身份认证协议在网络安全中占据十分重要的地位，对网络应用的安全有着非常重要的作用。

3.2.1 单向认证

- 单向认证是指通信双方中，只有一方对另一方进行认证。通常，单向认证协议包括三个步骤：应答方B通过网络发送一个挑战；发起方A回送一个对挑战的响应；应答方B检查此响应，然后再进行通信。单向认证既可以采用对称密码技术实现，也可以采用公钥密码技术实现。

基于对称加密的单向认证方案

- 在图3-1(a)所示协议中，B随机选择一个挑战R发送给A，A收到后使用共享的密钥K_{AB}加密R并将**加密结果**发送给B，则**B解密**得到R'，通过验证R=R'来实现对A的单向身份认证。
- 图3-1(b)所示协议是图3-1(a)的一个变形。B随机选择一个挑战R，并将R加密发送给A，A收到后使用共享的密钥K_{AB}解密收到的数据，得到R'并发送给B。同样，B可以验证R=R'来实现对A的认证。

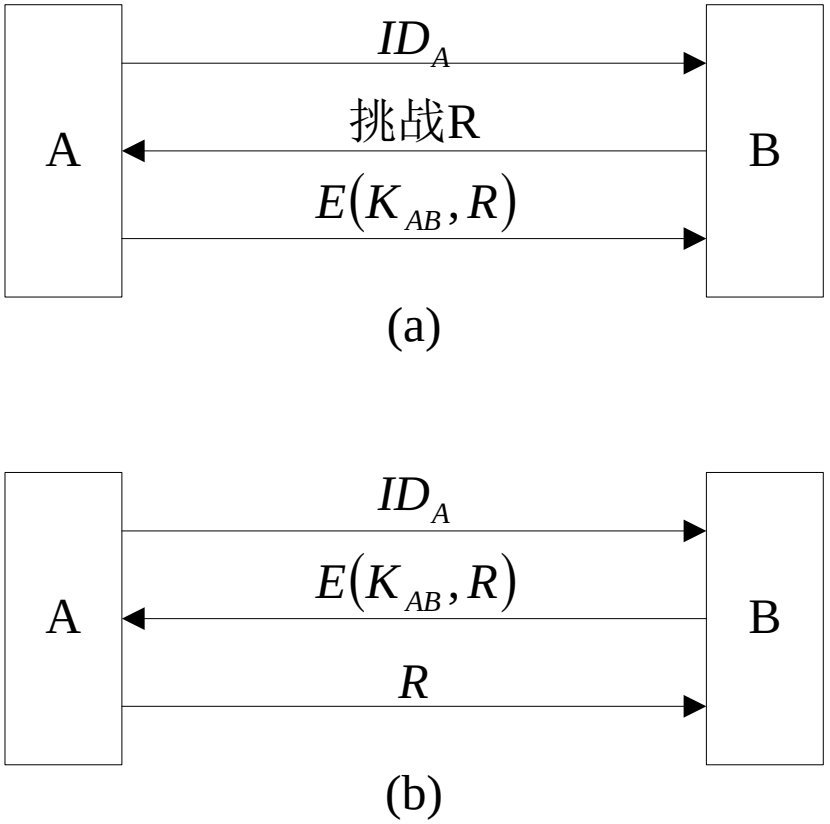


图3-1 基于对称加密的单向认证

基于对称加密的、KDC干预的单向认证

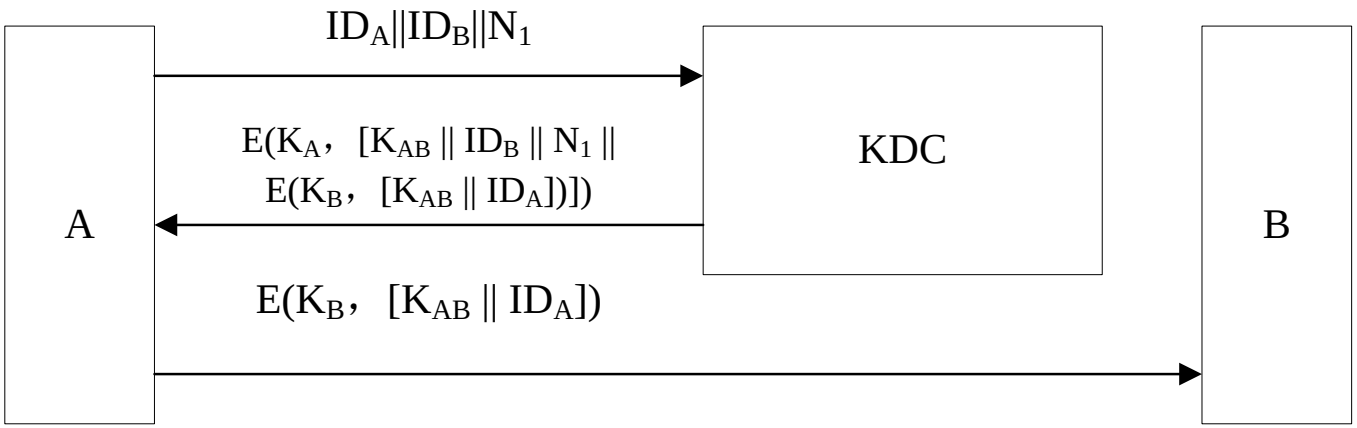


图3-2 基于对称加密的、KDC干预的单向认证

- 每个用户与密钥分配中心(KDC)共享唯一的主密钥，A有一个除了它外只有KDC知道的密钥K_A，同样，B有一个K_B。设A要与B建立一个逻辑连接，需要用一个一次性的会话密钥来保护数据的传输。

(1)A向KDC请求一个会话密钥以保护与B的逻辑连接。消息中有A和B的标识及唯一的标识N_1，这个标识我们称为临时交互号(nonce，不重复数)。

(2)KDC以用K_A加密的消息做出响应。消息中有两项内容是给A的。

- 一次性会话密钥K_{AB}，用于会话。
- 原始请求消息，包括临时交互号，以使A使用适当的请求匹配这个响应。

此外，消息中有两项内容是给B的。

- 一次性会话密钥K_{AB}，用于会话。
- A的标识符ID_A。

- 这两项用K_B(KDC与B共享的主密钥)加密。它们将发送给B，以建立连接并证明A的标识。

(3)A存下会话密钥备用，并将消息的后两项发给B，即$E(K_B, [K_{AB}\|ID_A])$。现在B已知道会话密钥K_{AB}，知道它稍后的通话伙伴是A(来自ID_A)，且知道这些消息来自于KDC(因为它是用K_B加密的)。至此，B实现了对A的认证过程。

基于公钥加密的简单单向认证方案

- 在图3-3(a)所示方案中，B给A发送一个挑战R，而A则用自己的私钥对R加密，B可以通过A的公钥解密并验证A的身份。
- 图3-3(b)中，B将挑战R用A的公钥加密后发送，A则用自己的私钥解密得到R'，B通过验证R=R'来实现对A的单向身份认证。

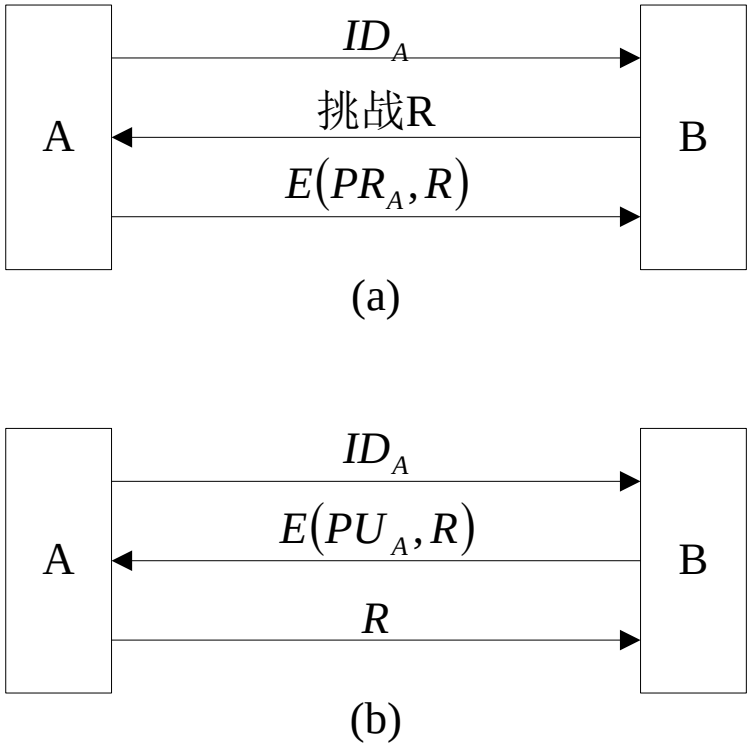


图3-3 基于公钥加密的单向认证

3.2.2 双向认证

- 双向认证是一个重要的应用领域，**指通信双方相互验证对方的身份**。
- 双向认证协议可以使通信双方确信对方的身份并交换会话密钥。保密性和及时性是认证的密钥交换中两个重要的问题。
- 为防止假冒和会话密钥的泄露，用户标识和会话密钥这样的重要信息必须以密文的形式传送，这就需要事先已有能用于这一目的的密钥或公钥。
- 因为可能存在消息重放，所以及时性非常重要。在最坏情况下，攻击者可以利用重放攻击威胁会话密钥或者成功地假冒另一方。

对付重放攻击的方法

- 对付重放攻击的方法之一是，在每个用于认证交换的消息后附加一个序列号，只有序列号正确的消息才能被接收。但是这种方法存在这样一个问题，即它要求每一通信方都要记录其他通信各方最后的序列号。因此，**认证和密钥交换一般不使用序列号，而是使用下列两种方法之一。**
- **时间戳：**仅当消息包含时间戳并且在A看来这个时间戳与其所认为的当前时间足够接近时，A才认为收到的消息是新消息，这种方法要求通信各方的时钟应保持同步。
- **挑战／应答：**若A要接收B发来的消息，则A首先给B发送一个临时交互号（挑战），并要求B发来的消息（应答）包含该临时交互号。

应对重放攻击的2种方法：适用场合

- **时间戳方法不适合于面向连接的应用**。第一，它需要某种协议保持通信各方的时钟同步，为了能够处理网络错误，该协议必须能够容错，并且还应能抗恶意攻击；第二，如果由于通信一方时钟机制出错而使同步失效，那么攻击成功的可能性就会增大；第三，由于各种不可预知的网络延时，不可能保持各分布时钟精确同步。因此，任何基于时间戳的程序都应有足够长的时限以适应网络延时，同时应有足够短的时限以使攻击的可能性最小。
- **另一方面，挑战／应答不适合于无连接的应用**，因为它要求在任何无连接传输之前必须先握手，这与无连接的主要特征相违背。

1.基于对称加密的双向认证

- 可以通过使用两层对称加密密钥的方式来保证分布式环境中通信的保密性。
- 通常，这种方法要使用一个可信的密钥分配中心(KDC)。在网络中，各方与KDC共享一个称为主密钥的密钥，KDC负责产生通信双方通信时短期使用的密钥（称为会话密钥），并用主密钥保护这些会话密钥的分配。基于KDC实现双向认证的经典协议是Needham和Schroeder设计的一个协议。

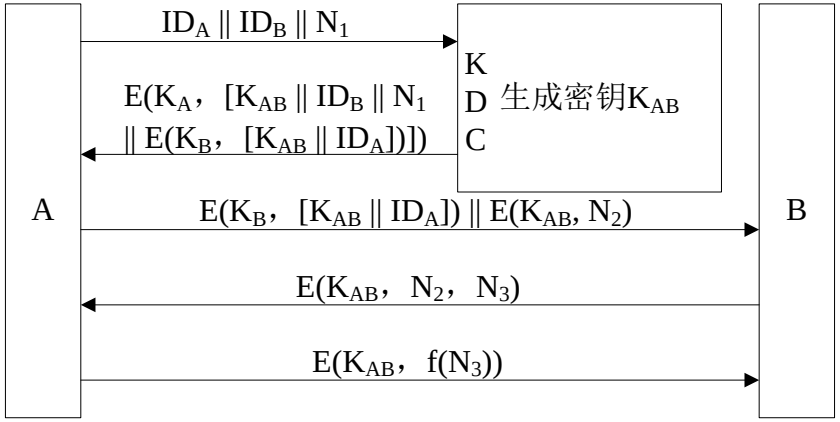


图3-4 基于对称加密的双向认证

基于对称加密的双向认证过程

(1)A→KDC: $ID_A||ID_B||N_1$

(2)KDC→A: $E(K_A,[K_{AB}||ID_B||N_1||E(K_B,[K_{AB}||ID_A])])$

(3)A→B: $E(K_B,[K_{AB}||ID_A])||E(K_{AB},N_2)$

(4)B→A: $E(K_{AB},N_2,N_3)$

(5)A→B: $E(K_{AB},f(N_3))$

- A、B分别和KDC共享密钥K_A和K_B，该协议的目的是要保证将会话密钥K_{AB}安全地分配给A和B。在步骤(1)，A首先告诉KDC，要和B通信。N_1的作用是防止攻击方通过消息重放假冒KDC;在步骤(2)，A安全地获得新的会话密钥K_{AB};在步骤(3)，A发送一个包括两个部分的消息给B：第一部分来自KDC，是用K_B加密的会话密钥K_{AB}和A的标识，第二部分是用K_{AB}加密的挑战N_2；在步骤(4)，B解密得到会话密钥K_{AB}和挑战N_2，然后B用K_{AB}加密N_2和新的挑战N_3，并发送给A。N_2的作用是证明B知道K_{AB}，N_3的作用是要求A证明自己知道K_{AB};步骤(5)使B确信A已知K_{AB}。至此，A和B相互认证了对方的身份，并且建立了会话密钥K_{AB}。

2.基于公钥加密的双向认证

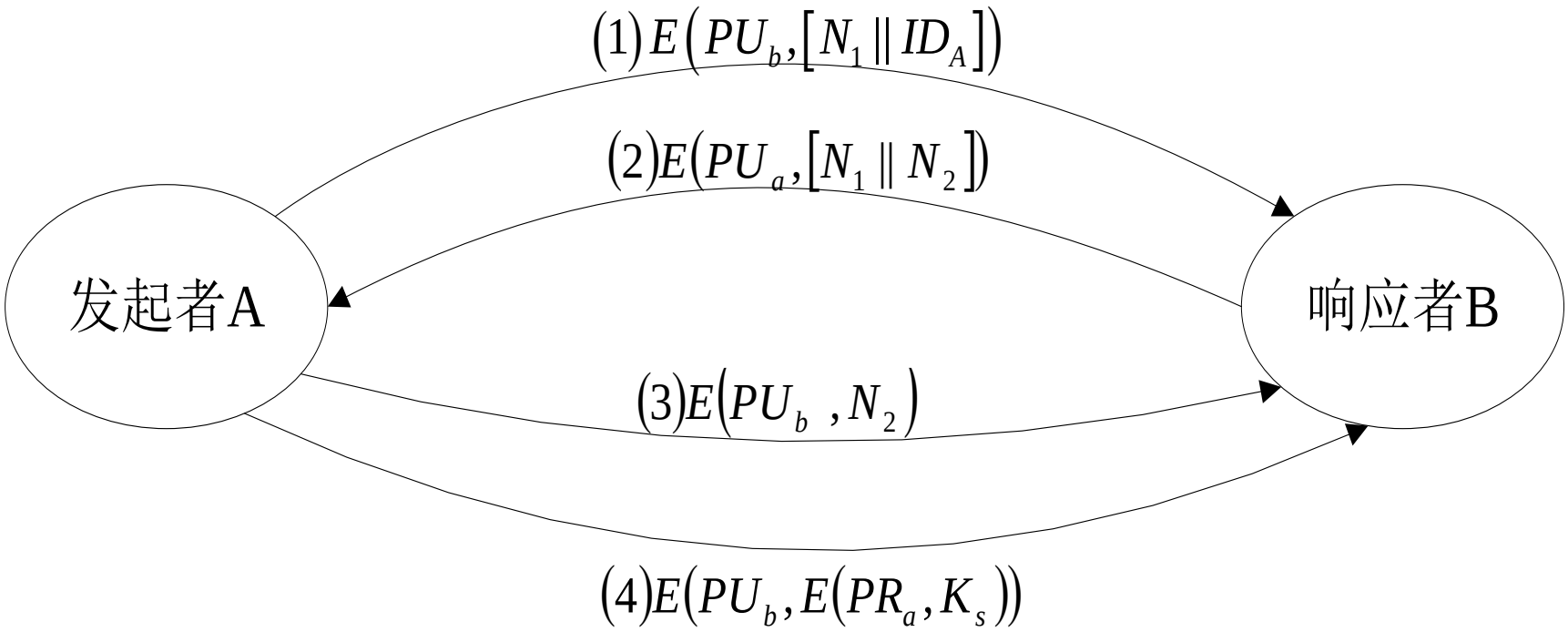


图3-5 基于公钥密码的双向认证

基于公钥密码的双向认证过程

① A用B的公钥对含有其标识ID_A和挑战(N_1)的消息加密，并发送给B。其中N_1用来唯一标识本次交易。

② B发送一条用PU_A加密的消息，该消息包含A的挑战(N_1)和B产生的新挑战(N_2)。因为只有B可以解密消息(1)，所以消息(2)中的N_1可使A确信其通信伙伴是B。

③ A用B的公钥对N_2加密，并返回给B，这样可使B确信其通信伙伴是A。至此，A与B实现了双向认证。

④ A选择密钥K_s，并将$M=E(PU_B, E(PR_A, K_s))$发送给B。使用B的公钥对消息加密可以保证只有B才能对它解密；使用A的私钥加密可以保证只有A才能发送该消息。

⑤ B用自己的私钥解密M得到$Y=E(PR_A, K_s)$，再用A的公钥解密Y得到密钥K_s。

3.3 Kerbores

- Kerberos是20世纪80年代美国麻省理工学院(MIT)开发的一种基于对称密码算法的网络认证协议，允许一个非安全的网络上的两台计算机通过交换加密消息互相证明身份。一旦身份得到验证，Kerberos协议给这两台计算机提供密钥，以进行安全的通信。
- Kerberos阐述了这样一个问题：假设有一个开放的分布环境，用户通过用户名和口令登录到工作站。从登录到登出这段时间称为一个登录会话。在某个登录过程中，用户可能希望通过网络访问各种远程资源，这些资源需要认证用户的身份。用户工作站替用户实施认证过程，以获得资源使用权，而用户不需知道认证的细节。服务器能够只对授权用户提供服务，并能鉴别服务请求的种类。

Kerberos与RFC

- Kerberos的设计目的就是解决分布式网络环境下，用户访问网络资源时的安全问题，即工作站的用户希望获得服务器上的服务，服务器能够对服务请求进行认证，并能限制授权用户的访问。
- Kerberos是为TCP/IP网络设计的可信第三方认证协议，利用可信第三方KDC(密钥分配中心)进行集中的认证。
- 目前常用的Kerberos有两个版本。版本4被广泛使用，而版本5改进了版本4中的安全性问题，并作为Internet标准草案(RFC 1510)。
- 2005年，RFC 1510被RFC 4120取代。

3.3.1 Kerberos版本4

- Kerberos通过提供一个集中的认证服务器来负责用户对服务器的认证和服务器对用户的认证。
- Kerberos的实现包括一个运行在网络上某个物理安全节点处的**密钥分发中心(key distribution center，KDC)以及一个函数库**，各需要认证用户身份的分布式应用程序调用这个函数库实现对用户的认证。
- Kerberos的设计目的是使用户通过用户名和口令登录到工作站，工作站基于口令生成密钥，并使用密钥和KDC联系，以代替用户获得远程资源的使用授权。

1.Kerberos配置

- Kerberos的版本4在协议中使用DES来提供认证服务。**每个实体都有自己的密钥，称为该实体的主密钥，这个主密钥是和KDC共享的**。
- 用户主密钥由用户口令生成，因此用户需要记住自己的口令，而网络设备则存储自己的主密钥。
- Kerberos服务器称为KDC，包括两个重要的模块：认证服务器(authentication server，AS)和门票授权服务器(TGS)。
- KDC有一个记录实体名字和相应主密钥的数据库。
- 为保证KDC数据库的安全，这些实体主密钥用KDC的主密钥加密。

用会话密钥增强安全性

- 用户通过用户名和口令登录到工作站，主密钥根据其口令生成。工作站可以记住用户名和口令，并使用这些信息来完成后面的认证过程。但是这样做不是很安全。如果用户在登录会话过程中运行了不可信软件，则易造成口令的泄露。
- 为了降低风险，在用户登录后，工作站首先向KDC申请一个会话密钥，而且只用于本次会话。随后，工作站忘掉用户名和口令，使用这个会话密钥和KDC联系，完成认证的过程，获得远程资源的使用授权。
- 会话密钥只在一段时间内有效，这大大降低了该密钥泄露造成安全问题的风险。

2.服务认证交换：获得会话密钥和TGT

- 在用户A登录工作站的时候，工作站向AS申请会话密钥。AS生成一个会话密钥S_A并用A的主密钥加密发送给A的工作站。此外，AS还发送一个门票授权门票(ticket-granting ticket，TGT)，TGT包含用KDC主密钥加密的会话密钥S_A、A的ID以及密钥过期时间等信息。A的工作站用A的主密钥解密，然后，工作站就可以忘记A的用户名和口令，而只需要记住S_A和TGT。每当用户申请一项新的服务，客户端则用TGT证明自己的身份，向TGS发出申请。

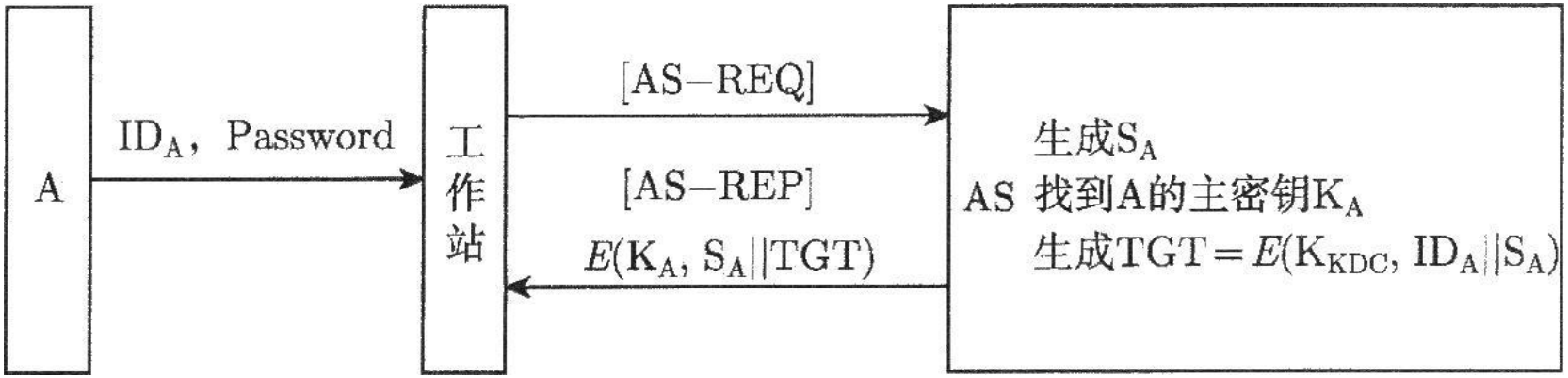


图3-6 获得会话密钥和TGT

获得会话密钥S_A和TGT的过程

- 用户A输入用户名和口令登录工作站，工作站以明文方式发送请求消息给KDC，消息中包括A的用户名。收到请求后，KDC(AS模块)使用A的主密钥加密访问TGS所需的证书，该证书包括：
  A. 会话密钥S_A。
  B. TGT。TGT包括会话密钥、用户名和过期时间，并用KDC的主密钥加密，因此，只有KDC才可以解密该TGT。
- 证书使用A的主密钥K_A加密，并发送给A的工作站。工作站将A的口令转换为DES密钥。工作站收到证书后，就用密钥解密证书，如果解密成功，则工作站抛弃A的主密钥K_A，只保留TGT和会话密钥。

获得会话密钥S_A和TGT的过程

- Kerberos中，将工作站发送给KDC的请求称为KRB_AS_REQ，即Kerberos认证服务请求(Kerberos authentication server request);将KDC的应答消息称为KRB_AS_REP，即Kerberos认证服务响应(Kerberos authentication server response)。消息交换过程可以简单描述为

(1) $A \rightarrow AS: ID_A || ID_{TGS} || TS_1$

(2) $AS \rightarrow A: E(K_A, [S_A || ID_{tgs} || TS_2 || Lifetime_2 || TGT])$

$$TGT = E(K_{KDC}, [S_A || ID_A || AD_A || ID_{tgs} || TS_2 || Lifetime_2])$$

表3-1 服务认证交换：获得TGT

KRB_AS_REQ：用户申请 TGT	
ID_A	告知 AS 客户端的用户标识
ID_{TGS}	告知 AS 用户请求访问 TGS
TS_1	使 AS 能验证客户端时钟是否与 AS 时钟同步
KRB_AS_REP：AS 返回 TGT	
K_A	基于用户口令的密钥便得 AS 与客户端能验证口令，保护消息的内容
S_{AS}	客户端可访问的会话密钥，由 AS 创建，使得客户端和 TGS 在不需要共享永久密钥的前提下安全交换信息
ID_{TGS}	标识该门票是为 TGS 生成的
TS_2	通知客户端门票发放的时间戳
$Lifetime_2$	通知客户端门票的生命期
TGT	客户端用于访问 TGS 的门票

3.服务授权门票交换：请求访问远程资源

- 有了TGT和会话密钥，A就可以与TGS通话。假设A请求访问远程服务器B的资源，由A向TGS发送消息，消息中包含TGT和所申请服务的标识ID。另外，此消息中还包含一个认证值，包括B的用户标识ID、网络地址和时间戳。与TGT的可重用性不同，此认证值仅能使用一次且生命期极短。Kerberos中将这个请求消息称为KRB_TGS_REQ。
- 当TGS接收到KRB_TGS_REQ消息后，**用K_{KDC}解密TGT**，TGT包含的信息说明用户A已得到会话密钥S_A，即相当于宣布“任何使用S_A的用户必为A”。接着，TGS使用该会话密钥解密认证消息，用得到的信息检查消息来源的网络地址。如匹配，则TGS确认该门票的发送者与门票的所有者是一致的，从而验证了A的身份。

服务授权门票交换：请求访问远程资源

- TGS为A与B生成一个共享密钥K_{AB}，并给A生成一个访问B的服务授权门票，门票的内容是使用B的主密钥加密的共享密钥K_{AB}和A的ID。A无法读取门票中的信息，因为门票用B的主密钥加密。为了获得B上的资源使用授权，A将门票发送给B，B可以解密该门票，获得会话密钥K_{AB}和A的ID。
- 然后，TGS给A发送一个应答消息，此消息称为KRB_TGS_REP，用TGS和A的共享会话密钥S_A加密。此应答消息内容包括A与服务器B的共享密钥K_{AB}、服务器B的标识ID以及A访问B的服务授权门票。

获得服务授权门票及消息交换过程

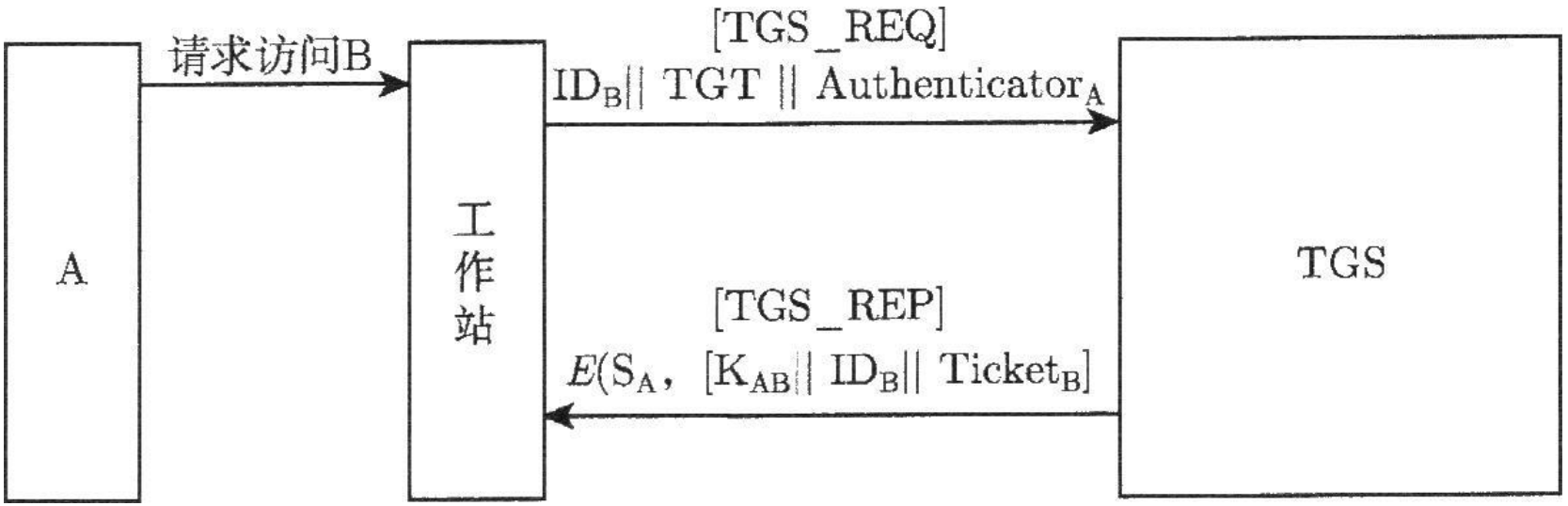


图3-7 获得服务授权门票

(3) A→TGS: $ID_B||$ TGT $||$ $Authenticator_C$

(4) TGS→A: $E(S_A, [K_{AB}|| ID_B|| TS_4|| Ticket_B])$

$$TGT = E(K_{KDC}, [S_A|| ID_A|| AD_A|| ID_{tgs}|| TS_2|| Lifetime_2])$$

$$Ticket_B = E(K_B, [K_{AB}|| ID_A|| AD_A|| ID_B|| TS_4|| Lifetime_4])$$

$$Authenticator_C = E(S_A, [ID_A|| AD_A|| TS_3])$$

KRB_TGS_REQ：客户端申请服务授权门票	
ID_B	告知 TGS 用户希望访问的服务器 B
TGT	告知 TGS 该用户已被 AS 认证
$Authenticator_C$	客户端生成的合法门票

KRB_TGS_REP：TGS 返回服务授权门票	
S_A	用 A 与 TGS 共享的密钥保护消息的内容
K_{AB}	客户端可访问的会话密钥，由 TGS 创建，使得客户端和服务器在不需要共享永久密钥的前提下安全交换信息
ID_B	标识该门票是为服务器 B 生成的
TS_4	通知客户端门票发放的时间戳
$Ticket_B$	客户端用于访问服务器 B 的门票
TGT	重用，以免用户重新输入口令
K_{KDC}	由 AS 和 TGS 共享的密钥加密的门票，防止伪造
S_A	TGS 可访问的会话密钥，用于解密认证消息，即认证门票
ID_A	标识门票的合法所有者
AD_A	防止门票在与申请门票时的不同工作站上使用
ID_{tgs}	向服务器确保门票解密正确
TS_2	通知 TGS 门票发放的时间
$Lifetime_2$	防止门票过期后继续使用
$Authenticator_A$	向 TGS 确保此门票的所有者与门票发放时的所有者相同，用短生命期防止重用
S_A	用客户端与 TCS 共享的密钥加密认证消息，防止伪造
ID_A	门票中必须与认证消息匹配的标识 ID
AD_A	门票中必须与认证消息匹配的网络地址
TS_3	通知 TGS 认证消息的生成时间

4.客户/服务器认证交换：访问远程资源

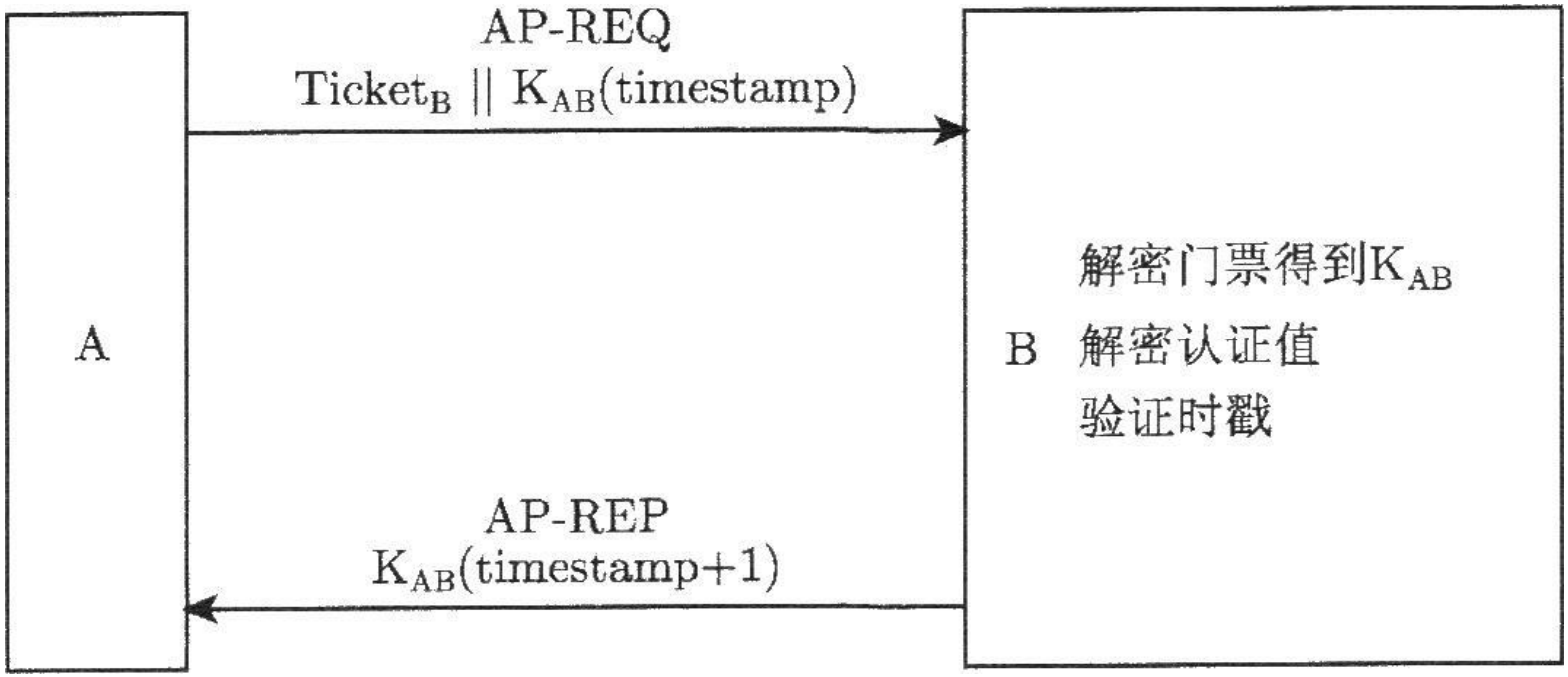


图3-8 访问远程资源

(5) A→B: $Ticket_B$|| $Authenticator_A$

(6) B→A: $E(K_{AB}, [TS_5+1])$(对所有认证)

$Ticket_B = E(K_B, [K_{AB}|| ID_A|| AD_A|| ID_B|| TS_4|| Lifetime_4])$

$Authenticator_A = E(K_{AB}, [ID_A|| AD_A|| TS_5])$

- 用户A的工作站给服务器B发送一个请求消息，此消息在Kerberos中称为KRB_AP_REQ，即“应用请求”消息。AP_REQ包含访问B的门票和认证值。认证值的形式是用A和B共享的会话密钥K_{AB}加密当前时间。
- B解密A发送的门票得到密钥K_{AB}和A的ID。然后，B解密认证值以确认和它通信的实体确实知道密钥，同时检查时间，以保证这个消息不是重放消息。现在，B已经认证了A的身份。B的应答消息在Kerberos中称为KRB_AP_REP。AP_REP的消息作用是为了实现A对B的认证。具体实现机制是B将解密得到的时间值加1，用K_{AB}加密后发回给A。A解密消息后可得到增加后的时间戳，由于消息是被会话密钥加密的，A可以确信此消息只可能由服务器B生成。消息中的内容确保该应答不是一个对以前消息的应答。
- 至此，客户端A与服务器B实现了双向认证，并共享一个密钥K_{AB}，该密钥可以用于加密在它们之间传递的消息或交换新的随机会话密钥。

表3-3客户 / 服务认证交换

KRB_AP_REQ：客户端申请服务	
$Ticket_B$	向服务器证明该用户通过了 AS 的认证
$Authenticator_B$	客户端生成的合法门票
KRB_AP_REP：可选的客户端认证服务器	
K_{AB}	向客户端 A 证明该消息来源于服务器 B
TS_5+1	向客户端 A 证明该应答不是对原来消息的应答
$Ticket_B$	可重用，使得用户在多次使用同一服务器时不需要向 TGS 申请新门票
K_B	用 TGS 与服务器共享的密钥加密的门票，防止仿造
K_{AB}	客户端可访问的会话密钥，用于解密认证消息
ID_A	标识门票的合法所有者
AD_A	防止门票在与申请门票时的不同工作站上使用
ID_B	确保服务器能正确解密门票
TS_4	通知服务器门票发放的时间
$Lifetime_4$	防止门票超时使用
$Authenticator_A$	向服务器确保此门票的所有者与门票发放时的所有者相同，用短生命期防止重用
K_{AB}	用客户端与服务器共享的密钥加密的认证消息，防止假冒
ID_A	门票中必须与认证消息匹配的标识 ID
AD_A	门票中必须与认证消息匹配的网络地址
TS_5	通知服务器认证消息的生成时间

3.3.2 Kerberos版本5

- Neuman and Kohl published version 5 in 1993 with the intention of **overcoming existing limitations and security problems**. Version 5 appeared as RFC 1510, and was made obsolete by RFC 4120 in 2005.
- In 2005, the Internet Engineering Task Force (IETF) Kerberos working group updated specifications. Updates included:
  - Encryption and Checksum Specifications (RFC 3961).
  - Advanced Encryption Standard (AES) Encryption for Kerberos 5 (RFC 3962).
  - A new edition of the Kerberos V5 specification "The Kerberos Network Authentication Service (V5)" (RFC 4120). This version obsoletes RFC 1510, clarifies aspects of the protocol and intended use in a more detailed and clearer explanation.
  - A new edition of the Generic Security Services Application Program Interface (GSS-API) specification "The Kerberos Version 5 Generic Security Service Application Program Interface (GSS-API) Mechanism: Version 2." (RFC 4121).
- MIT makes an implementation of Kerberos freely available, under copyright permissions similar to those used for BSD.

1.版本改进

- Kerberos版本5对版本4存在的一些缺陷进行了改进。

1)加密系统依赖性

- 版本4使用DES，目前(2019年)DES已经不安全。**版本5**用加密类型标记密文，使得它**可以使用任何加密技术**。加密密钥也加上类型和长度标记，允许不同的算法使用相同的密钥。

2)Internet协议依赖性

- 版本4需要使用IP地址，不支持其他地址类型。版本5用类型和长度标记网络地址，**允许使用任何类型的网络地址。**

1.版本改进

3)消息字节顺序

- 版本4中，由消息的发送者用标记说明规定消息的字节顺序，而不遵循已有的惯例。在版本5中，所有消息结都遵循**抽象语法表示(ASN.1)**和**基本编码规则(BER)**的规定，提供一个明确无二义的消息字节顺序。

4)门票的生命期

- 版本4中，门票的生命期用一个8位表示，每单位代表5分钟。因此，最大生命期为28*5=1280分钟，约为21小时。这对某些应用可能不够长。在版本5中，门票中包含了精确的起始时间和终止时间，允许门票拥有**任意长度的生命期**。

1.版本改进

5)向前认证

- 版本4中，不允许发给一个客户端的证书被转发到其他主机或被其他用户进行其他相关操作。此操作是指服务器为了完成客户端请求的服务而请求其他服务器协作。版本5提供这项功能。

6)域间认证

- 版本4中，N个域的互操作需要N^2个Kerberos-to-Kerberos关系。版本5中支持一种需要较少连接的方法。

7)冗余加密

- 版本4中，对提供给客户端的门票进行两次加密，第一次使用的是目标服务器的密钥，第二次使用的是客户端密钥。版本5取消了第二次加密，即用用户密钥进行的加密，因为第二次加密并不是必须的。

1.版本改进

8)PCBC加密

- 版本4加密使用DES的非标准模式PCBC，此种模式已被证明易受交换密码块攻击。版本5提供了精确的完整性检查机制，并能够用标准的CBC模式加密。

9)会话密钥

- 版本4中，每张门票中包含一个会话密钥，此门票被多次用来访问同一服务器，因而可能遭受重放攻击。在版本5中，客户端与服务器可以协商一个用于特定连接的子会话密钥，每个子会话密钥仅被使用一次。这种新的客户端访问方式将会减少重放攻击的机会。

Kerberos版本5消息交换

表3-4 Kerberos版本5消息交换

(a) 认证服务交换：获取门票授权门票	
(1) A→AS	$\text{Options} \| \text{ID}_A \| \text{Realm}_A \| \text{ID}_{tgs} \| \text{Times} \| \text{Nonce}_1$
(2) AS→A	$\text{Realm}_A \| \text{ID}_A \| \text{Ticket}_{tgs} \| E(K_A, [S_A \| \text{Times} \| \text{Nonce}_1 \| \text{Realm}_{tgs} \| \text{ID}_{tgs}])$
	$\text{Ticket}_{tgs} = E(K_{tgs}, [\text{Flags} \| S_A \| \text{Realm}_A \| \text{ID}_A \| \text{AD}_A \| \text{Times}])$
(b) 服务授权门票交换：获取服务授权门票	
(3) A→TGS	$\text{Options} \| \text{ID}_B \| \text{Times} \| \text{Nonce}_2 \| \text{Ticket}_{tgs} \| \text{Authenticator}_A$
(4) TGS→A	$\text{Realm}_A \| \text{ID}_A \| \text{Ticket}_B \| E(S_A, [K_{AB} \| \text{Times} \| \text{Nonce}_2 \| \text{Realm}_B \| \text{ID}_B])$
	$\text{Ticket}_{tgs} = E(K_{tgs}, [\text{Flags} \| S_A \| \text{Realm}_A \| \text{ID}_A \| \text{AD}_A \| \text{Times}])$
	$\text{Ticket}_B = E(K_B, [\text{Flags} \| K_{AB} \| \text{Realm}_B \| \text{ID}_A \| \text{AD}_A \| \text{Times}])$
	$\text{Authenticator}_A = E(S_A, [\text{ID}_A \| \text{Realm}_A \| \text{TS}_1])$
(c) 客户/服务器认证交换：获取服务	
(5) A→B	$\text{Options} \| \text{Ticket}_B \| \text{Authenticator}_A$
(6) B→A	$E(K_{AB}, [\text{TS}_2 \| \text{Subkey} \| \text{Seq\#}]$
	$\text{Ticket}_B = E(K_B, [\text{Flags} \| K_{AB} \| \text{Realm}_A \| \text{ID}_A \| \text{AD}_A \| \text{Times}])$
	$\text{Authenticator}_A = E(K_{AB}, [\text{ID}_A \| \text{Realm}_A \| \text{TS}_2 \| \text{Subkey} \| \text{Seq\#}])$

(a)认证服务交换：获取门票授权门票

- 消息(1)是客户端请求门票授权门票过程。如前所述，它包括用户和TGS的标识，新增的元素包括：
  - ✓1)Realm：标识用户所属的域。
  - ✓2)Options：用于请求在返回的门票中设置指定的标识位，如下所述。
  - ✓3)Times：用于客户端请求在门票中设置时间。
    - from，请求门票的起始时间
    - till，请求门票的过期时间
    - rtime，请求till更新时间
  - ✓4)Nonce：在消息(2)中重复使用的临时交互号，用于确保应答是刷新的，且未被攻击者使用。

(a)认证服务交换：获取门票授权门票

- 消息(2)返回门票授权门票，标识客户端信息和一个用用户口令形成的密钥加密的数据块。
- 该数据块包含客户端和TGS间使用的会话密钥，消息(1)中设定的时间和临时交互号以及TGS的标识信息。
- 门票本身包含会话密钥、客户端的标识信息、需要的时间值、影响门票状态的标志和选项。

- 这些标志为版本5带来的一些新功能将在以后讨论。

(b) 服务授权门票交换：获取服务授权门票

- **版本4和版本5的服务授权门票交换：**

- 两者的消息(3)均包含认证码、门票和请求服务的名字。在版本5中，还包括与消息(1)类似的门票请求的时间、选项和一个临时交互号；认证码的作用与版本4中相同。

- 消息(4)与消息(2)结构相同，返回门票和一些客户端需要的信息，后者被客户端和TGS共享的会话密钥加密。

(c)服务授权门票交换：获取服务

- 版本5对**客户／服务器认证交换**进行了一些改进，如在消息(5)中，客户端可以请求选择双向认证选项。认证也增加了以下新域。
  - ✓Subkey：客户端选择一个子密钥保护某一特定应用会话，如果此域被忽略，则使用门票中的会话密钥$K_{c,v}$。
  - ✓Sequence(序号)：可选域，用于说明在此次会话中服务器向客户端发送消息的序列号。将消息排序可以防止重放攻击。
- 如果请求双向认证，则服务器按消息(6)应答。该消息中包含从认证消息中得到的时间戳。在版本4中，该时间戳被加1，而在版本5中，由于攻击者不可能在不知道正确密钥的情况下创建消息(6)，因此，不需要对时间戳进行上述处理。如果有子密钥域存在，则覆盖消息(5)中相应的子密钥域。而选项序列号则说明了客户端使用的起始序列号。

2.门票标志

表3-5 Kerberos版本5标志

INITIAL	按照 AS 协议发布的服务授权门票，而不是基于门票授权门票发布的
PRE-AUTHENT	在初始认证中，客户在授予门票前即被 KDC 认证
HW-AUTHENT	初始认证协议要求使用带名客户端独占硬件资源
RENEWABLE	告知 TGS 此门票可用于获得最近超时门票的新门票
MAY-POSTDATE	告知 TGS 事后通知的门票可能基于门票授权门票
POSTDATED	表示该门票是事后通知的，终端服务器可以检查 authtime 域，查看认证发生的时间
INVALID	不合法的门票在使用前必须通过 KDC 使之合法化
PROXIABLE	告知 TGS 根据当前门票可以发放给不同网络地址新的服务授权门票
PROXY	表示该门票是一个代理
FORWARDABLE	告知 TGS 根据此门票授权门票可以发放给不同网络地址新的门票授权门票
FORWARDED	表示该门票或是经过转发的门票或是基于转发的门票授权门票认证后发放的门票

Kerberos版本5标志

- **标志INITIAL：**用于表示门票是由AS发放的，而不是由TGS发放的。当客户端向TGS申请服务授权门票时，必须拥有AS发放的门票授权门票。
- **标志PRE-AUTHENT：**如果被设置，则表示当AS接收初始请求消息(1)时，在发放门票前应先对客户端进行认证，其预认证的确切格式在此未做详细说明。
- **标志RENEWABLE：**一个具有标志RENEWABLE的门票中包含两个有效期：一个是此特定门票的有效期，另一个是最大许可值的有效期。客户端可以通过将门票提交给TGS，申请得到新的有效期的方法获得新门票。

Kerberos版本5标志

- **标志MAY-POSTDATE：**客户端可请求AS提供一个具有标志MAY-POSTDATE的门票授权门票。客户端可以使用此门票从TGS申请一个具有标志POSTDATED或INVALID的门票，然后，客户端提交合法的超时门票。
- **PROXIABLE标志：**如果客户端想使用代理机制，需要申请获得一个带PROXIABLE标志的门票授权门票。当此门票传给TGS时，TGS发布一个具有不同网络地址的服务授权门票。该门票的标志PROXY被设置，接收到这种门票的应用可以接收它或请求进一步认证，以提供审计跟踪。
- **FORWARDED标志：**如果门票被设置为FORWARDABLE，TGS给申请者发放一个具有不同网址和FORWARDED标志的门票授权门票，此门票可以被送往远程TGS。这使得用户端在不需要每个Kerberos都包含与其他各不同域中的Kerberos共享密钥的前提下，可访问不同域的服务器。

3.4 PKI技术

- PKI：Public Key Infrastructure，公钥基础设施
- PKI是一种遵循标准的、利用公钥加密技术的一套安全基础平台的技术和规范。

3.4.1 PKI体系结构

- 简单的说，PKI是基于公钥密码技术，支持公钥管理，提供真实性、保密性、完整性以及可追究性安全服务，具有普适性的安全基础设施。PKI的核心技术围绕建立在公钥密码算法之上的数字证书的申请、颁发、使用与撤销等整个生命周期进行展开，主要目的就是用来安全、便捷、高效地分发公钥，为用户建立一个安全的网络环境，保证网络上信息的安全传输。

PKI应用系统的组成

- IETF的PKI小组制订了一系列的协议，定义了基于X.509证书的PKI模型框架，称为PKIX。PKIX系列协议定义了证书在Internet上的使用方式，包括证书的生成、发布、获取，各种密钥产生和分发的机制，以及实现这些协议的轮廓结构。狭义的PKI一般指PKIX。

- 个完整的PKI应用系统必须具有权威认证机构(CA)、数字证书库、密钥备份及恢复系统、证书作废系统、应用接口(API)等基本构成部分，如图3-9所示。
- 构建PKI也将围绕着这五大关键元素来着手构建。

PKI应用系统的组成

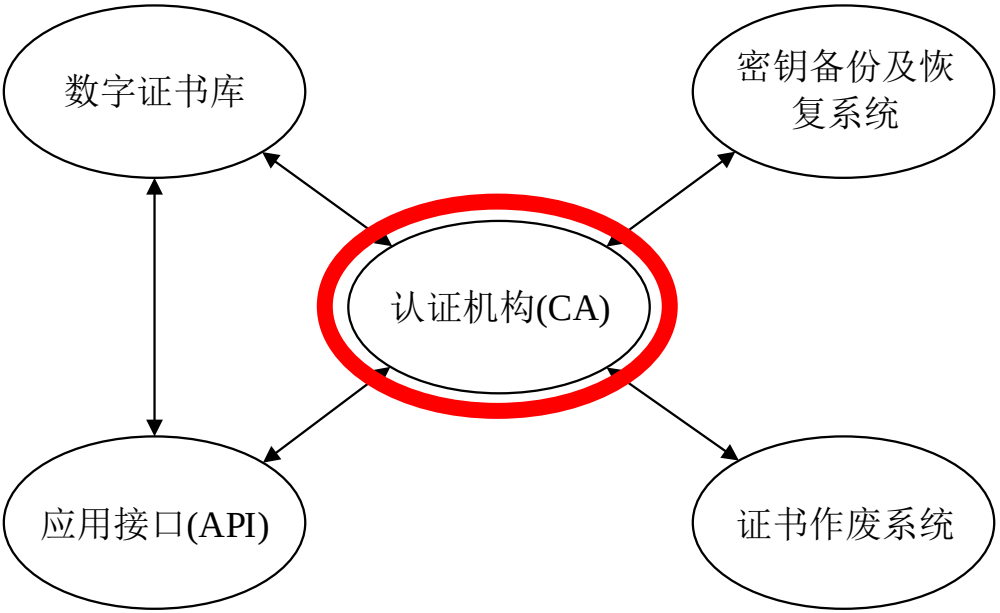


图3-9 PKI体系结构

- **认证机构(CA)：**CA是PKI的核心执行机构，是PKI的主要组成部分，人们通常称它为认证中心。CA是数字证书生成、发放的运行实体，在一般情况下也是证书撤销列表(CRL)的发布点，在其上常常运行着一个或多个注册机构(RA)。
- CA必须具备权威性的特征。

PKI应用系统的组成

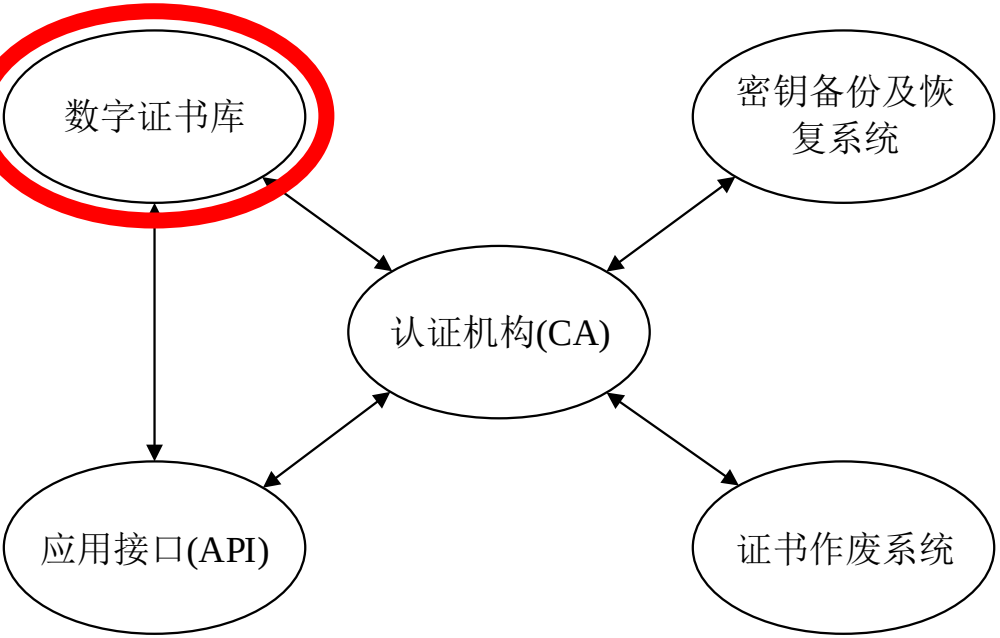


图3-9 PKI体系结构

- **数字证书库：**证书库是CA颁发证书和撤销证书的集中存放地，可供公众进行开放式查询。一般来说，查询的目的有两个：

① 其一是想得到与之通信实体的公钥；

② 其二是要验证通信对方的证书是否已进入“黑名单”。

- 证书库还提供了存取证书撤销列表(CRL)的方法。
- 目前广泛使用的是X.509证书。

PKI应用系统的组成

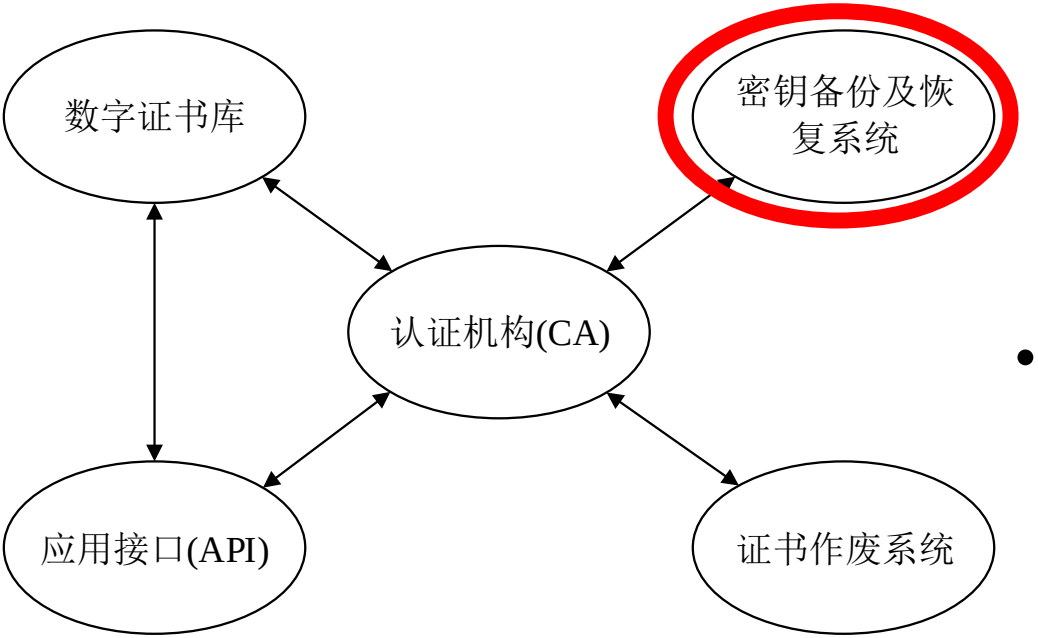


图3-9 PKI体系结构

- **密钥备份及恢复系统：**如果用户丢失了用于解密数据的密钥，则数据将无法被解密，这将造成合法数据丢失。
- 为避免这种情况，PKI提供备份与恢复密钥的机制。但是密钥的备份与恢复必须由可信的机构来完成。
- 密钥备份与恢复只能针对解密密钥，**签名私钥为确保其唯一性而不能够作备份。**

PKI应用系统的组成

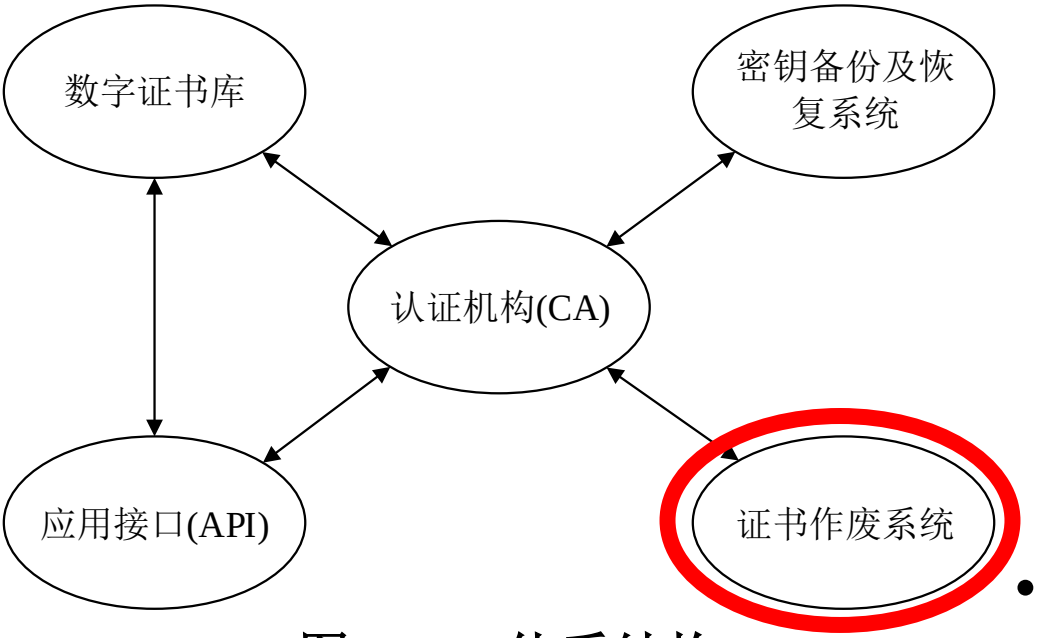


图3-9 PKI体系结构

- **证书作废系统：**证书作废处理系统是PKI的一个必备的组件。证书有效期以内也可能需要作废，原因可能是密钥介质丢失或用户身份变更等。在PKI体系中，作废证书一般通过将证书列入作废证书表(CRL)来完成。
- 通常，系统中由CA负责创建并维护一张及时更新的CRL，而由用户在验证证书时负责检查该证书是否在CRL之列。

PKI应用系统的组成

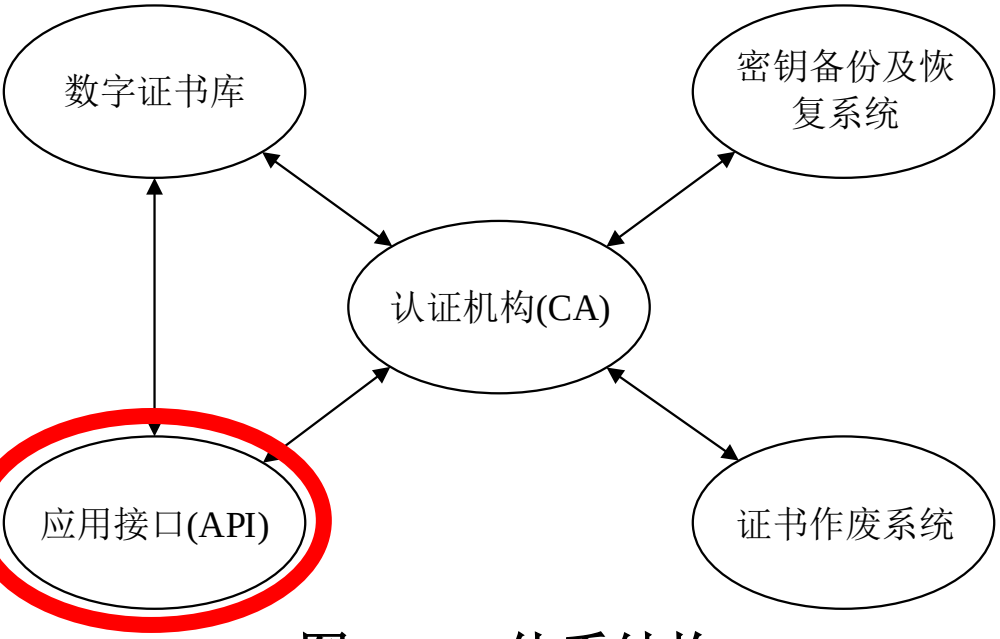


图3-9 PKI体系结构

- **应用接口(API)：** PKI的价值在于使用户能够方便地使用加密、数字签名等安全服务，因此，一个完整的PKI必须提供良好的应用接口系统，使得各种各样的应用能够以安全、一致、可信的方式与PKI交互，确保安全网络环境的完整性和易用性。

3.4.2 X.509数字证书

PKI中广泛使用了X.509证书。一个X.509证书包含以下信息：

- **版本号(Version)：**区分合法证书的不同版本。目前定义了三个版本，版本1的编号为0，版本2的编号为1，版本3的编号为2。
- **序列号(Serial number)：**一个整数，和签发该证书的CA名称一起惟一标识该证书。
- **签名算法标识(Signature algorithm identifier)：**指定证书中计算签名的算法，包括一个用来识别算法的子域和算法的可选参数。

X.509数字证书

- **签发者(Issuer name)：**创建、签名该证书的CA的X.500格式名字。
- **有效期(Period of validity)：**包含两个日期，即证书的生效日期和终止日期。
- **证书主体名(Subject name)：**持有证书的主体的X.500格式名字，证明此主体是公钥的所有者。
- **证书主体的公钥信息(Subject's public-key information)：**主体的公钥以及将被使用的算法标识，带有相关的参数。

X.509数字证书

- **签发者惟一标识(Issuer unique identifier)：**版本2和版本3中可选的域，用于惟一标识认证中心CA。
- **证书主体惟一标识(Subject unique identifier)：**版本2和版本3中可选的域，用于惟一标识证书主体。
- **扩展(Extensions)：**仅仅出现在版本3中，一个或多个扩展域集。
- **签名(Signature)：**覆盖证书的所有其他域，以及其他域被CA私钥加密后的散列代码，以及签名算法标识。
- CA用它的私钥对证书签名，如果用户知道相应的公钥，则用户可以验证CA签名证书的合法性，这是一种典型的数字签名方法。

3.4.3 认证机构

- PKI系统的关键是实现对公钥密码体制中公钥的管理。认证机构CA便是一个能够提供相关证明的机构。CA是基于PKI进行网上安全活动的关键，主要负责产生、分配并管理参与活动的所有实体所需的数字证书，其功能类似于办理身份证、护照等证件的权威发证机关。CA必须是各行业、各部门及公众共同信任并认可的、权威的、不参与交易的第三方网上身份认证机构。
- 在PKI系统中，CA管理公钥的整个生命周期，其功能包括签发证书、规定证书的有效期限，同时在证书发布后，还要负责对证书进行撤销、更新和归档等操作。从证书管理的角度，每一个CA的功能都是有限的，需要按照上级CA的策略，负责具体的用户公钥的签发、生成和发布，以及CRL的生成和发布等职能。

CA的主要职能

① 制定并发布本地CA策略。但本地策略只是对上级CA策略的补充，而不能违背。
② 对下属各成员进行身份认证和鉴别。
③ 发布本CA的证书，或者代替上级CA发布证书。
④ 产生和管理下属成员的证书。
⑤ 证实RA的证书申请，返回证书制作的确认信息，或返回已制作的证书。
⑥ 接收和认证对所签发证书的撤销申请。
⑦ 产生和发布所签发证书和CRL。
⑧ 保存证书、CRL信息、审计信息和所制定的策略。

典型CA的构成

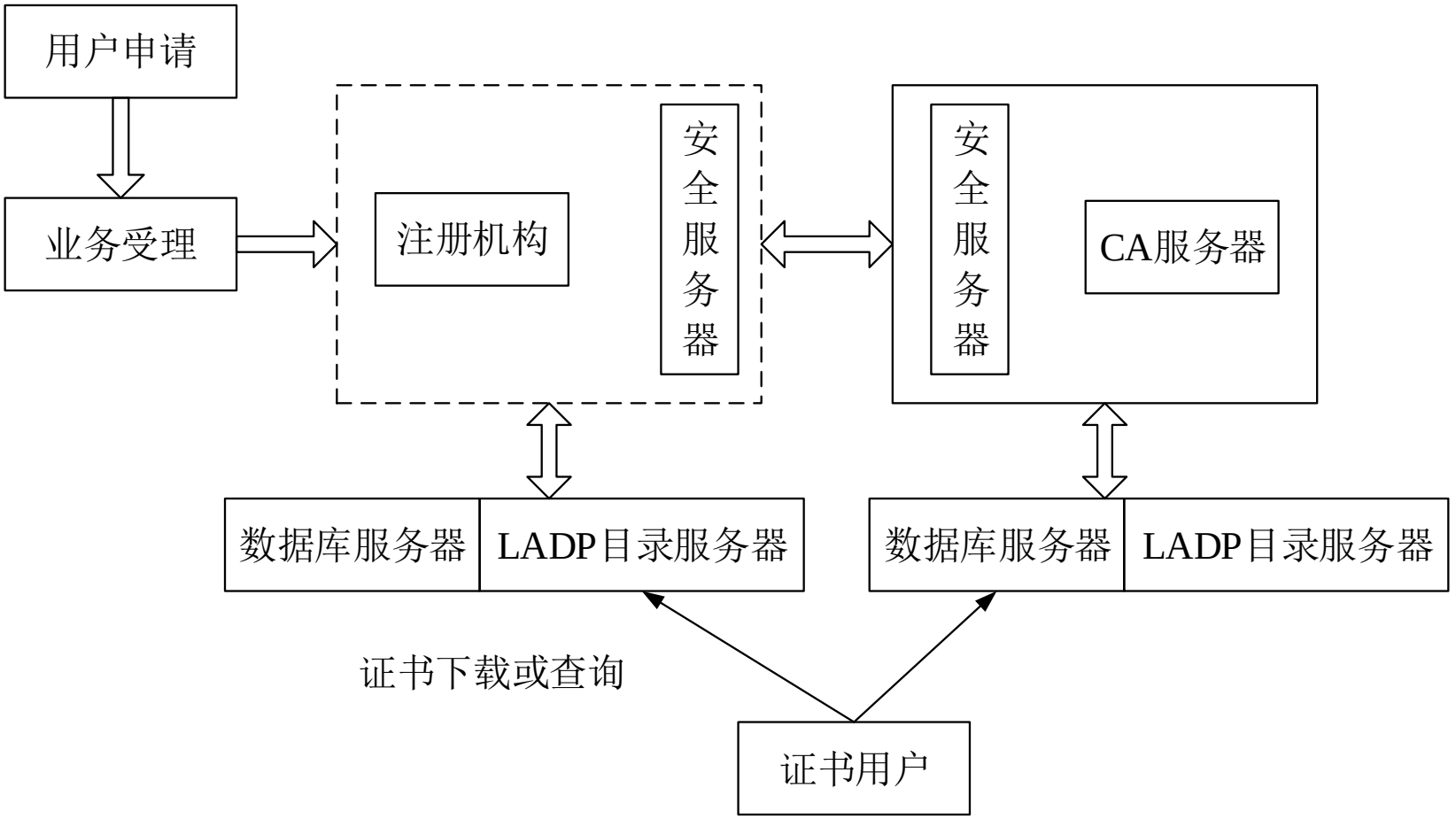


图3-10 典型CA的构成

3.4.4 PKIX相关协议

1) PKIX基础协议

- PKIX的基础协议以RFC2459和RFC3280为核心，定义了X.509 v3公钥证书和X.509 v2 CRL的格式、数据结构和操作等，用以保证PKI基本功能的实现。
- 此外，PKIX还在RFC2528、RFC3039、RFC3279等的基础上定义了基于X.509 v3的相关算法和格式等，以加强X.509 v3公钥证书和X.509 v2 CRL在各应用系统之间的通用性。

2) PKIX管理协议

- PKIX体系中定义了一系列的操作，它们是在管理协议的支持下进行工作的。管理协议主要完成以下的任务：

① 用户注册

② 用户初始化

③ 认证

④ 密钥对的备份和恢复

⑤ 自动的密钥对更新

⑥ 证书撤销请求

⑦ 交叉认证

3) PKIX安全服务和权限管理的相关协议

- PKIX中安全服务和权限管理的相关协议主要是进一步完善和扩展PKI安全架构的功能，通过RFC3029、RFC3161、RFC3281等定义。
- 在PKIX中，**不可抵赖性**通过数字时间戳DTS(digital timestamp)和数据有效性验证服务器DVCS(data validation and certification server)实现。
- 在CA/RA中使用的DTS，是对时间信息的数字签名，主要用于确定在某一时间某个文件确实存在或者确定多个文件的时间上的逻辑关系，是实现不可抵赖性服务的核心。
- DVCS的作用则是验证签名文档、公钥证书或数据存在的有效性，其验证声明称为**数据有效性证书**。DVCS是一个可信第三方，是用来实现不可抵赖性服务的一部分。权限管理通过属性证书来实现。属性证书利用属性和属性值来定义每个证书主体的角色、权限等信息。

3.4.5 PKI信任模型

- 实体A信任B，即A假定实体B严格地按A所期望的那样行动。如果一个实体认为CA能够建立并维持一个准确地对公钥属性的绑定，则它信任该CA。
- 所谓**信任模型，就是提供用户双方相互信任机制的框架**，是PKI系统整个网络结构的基础。
- 信任模型主要明确回答了以下几个问题：
  ① 一个PKI用户能够信任的证书是怎样被确定的?
  ② 这种信任是怎样建立的?
  ③ 在一定的环境下，这种信任如何被控制?

1.层次模型

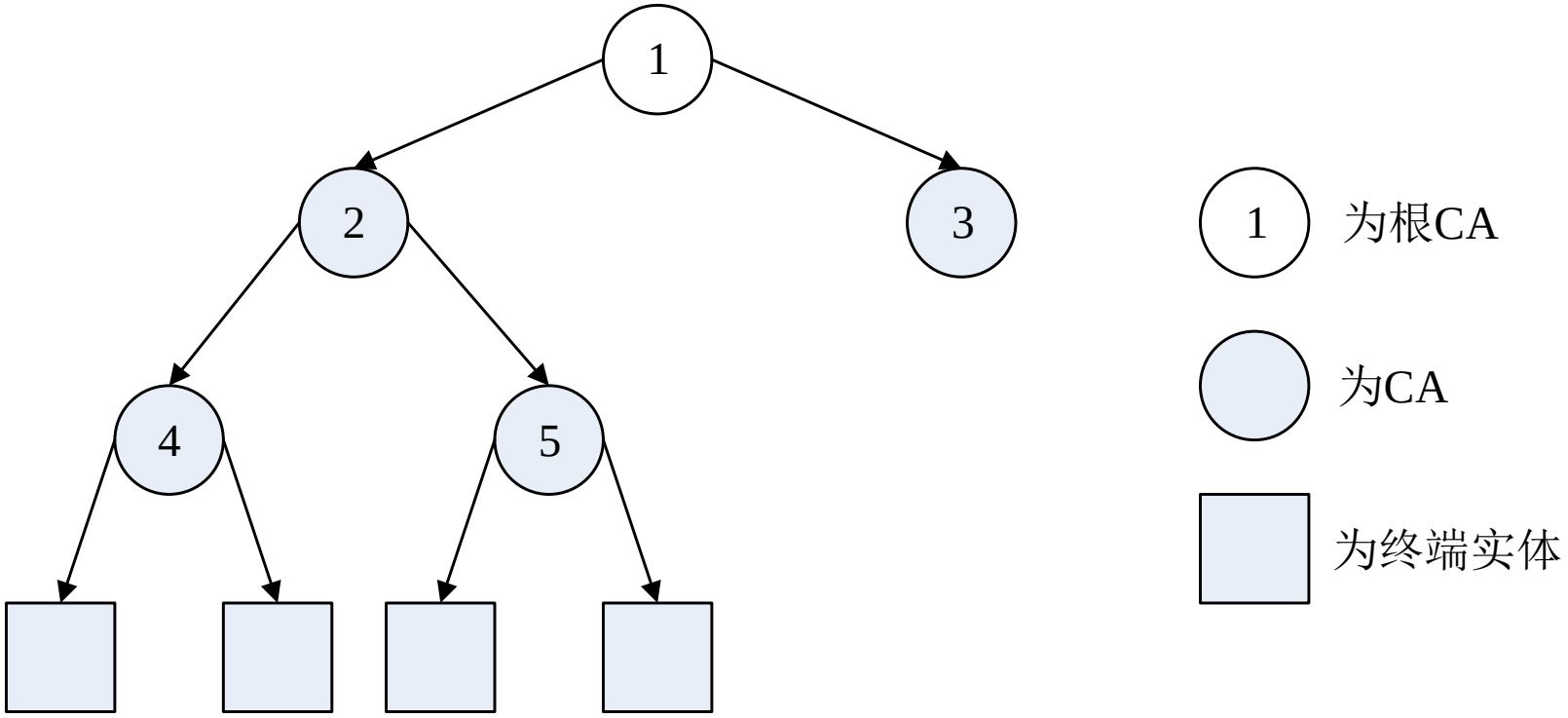


图3-12 层次模型

2.交叉模型

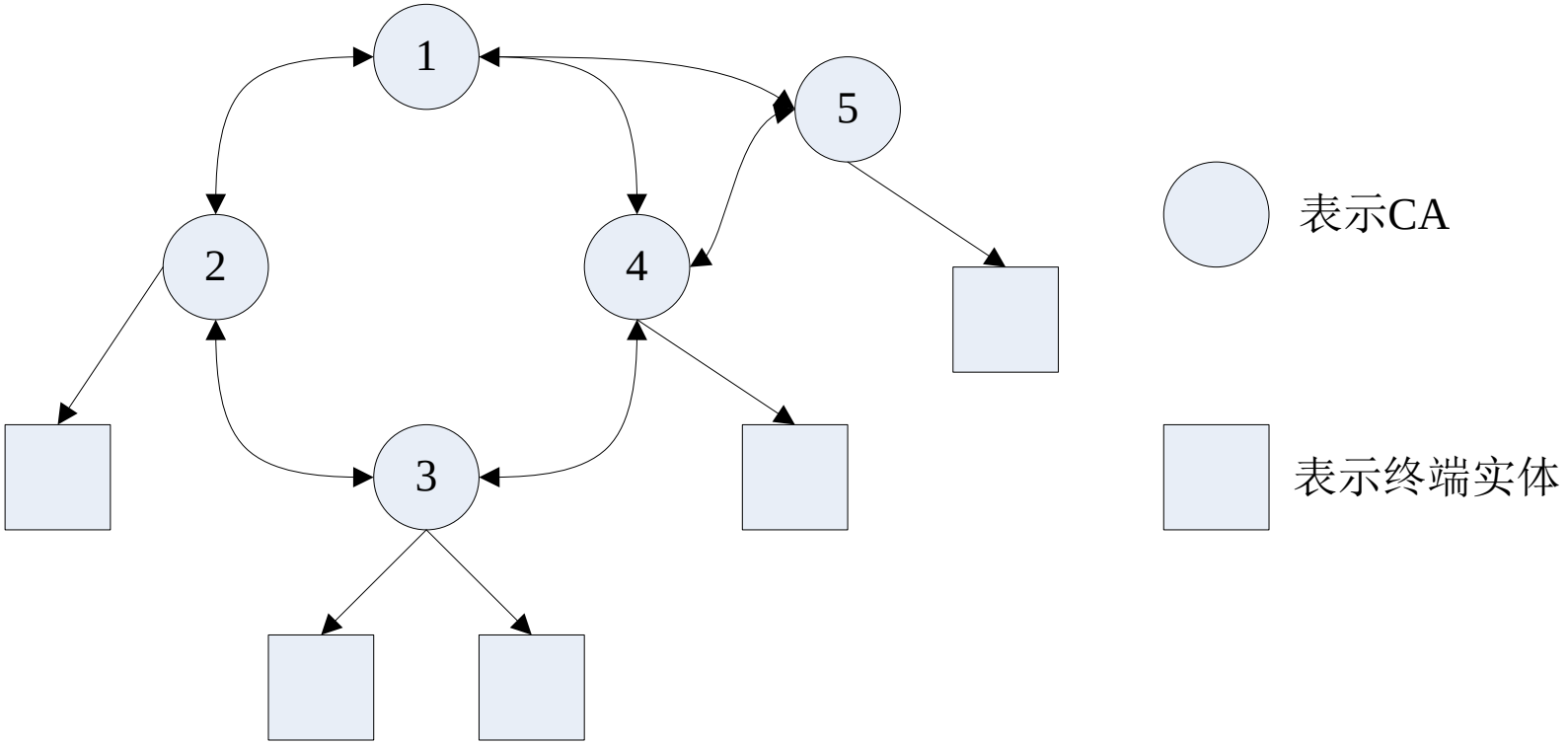


图3-12 交叉模型

3.混合模型

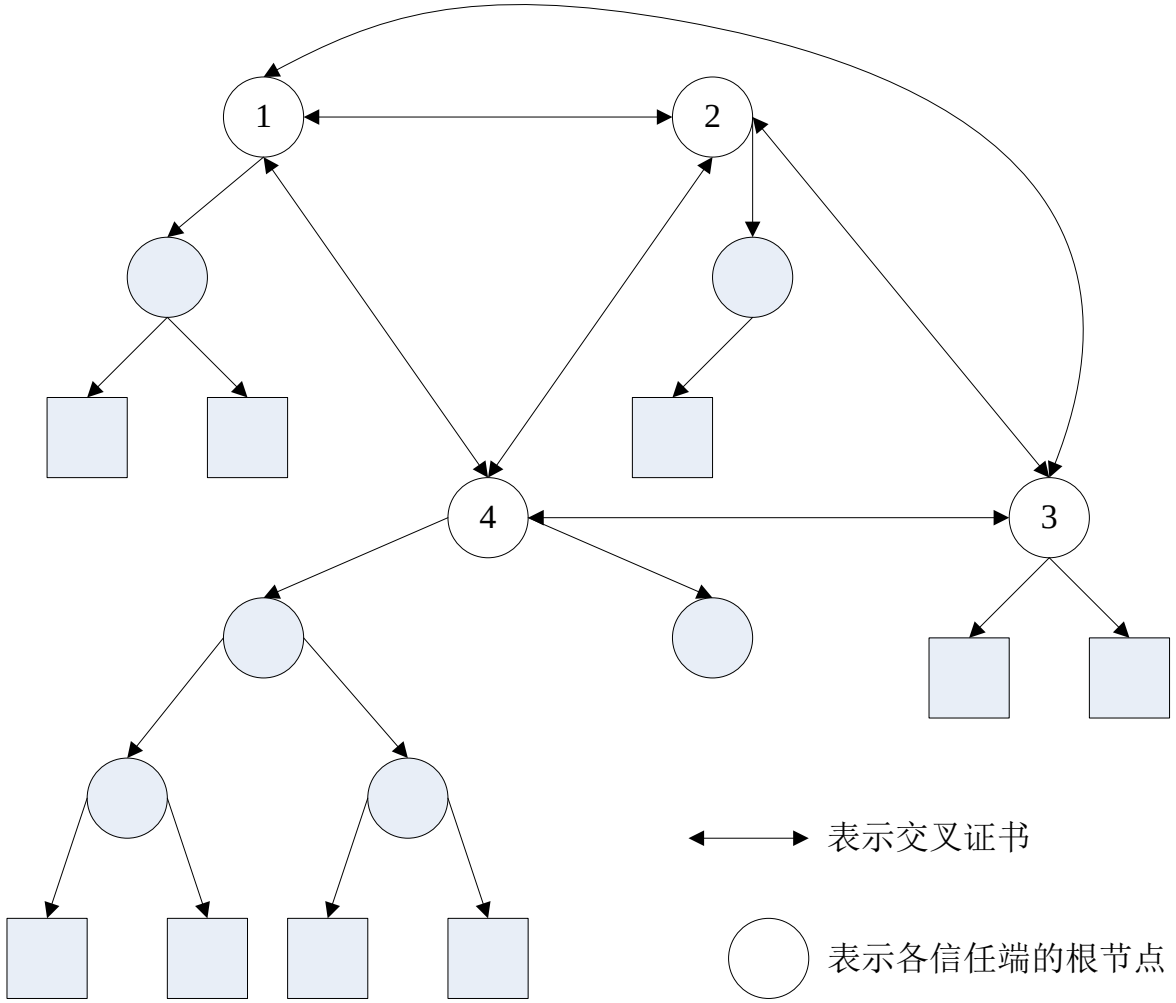


图3-13 混合模型

4.桥CA模型

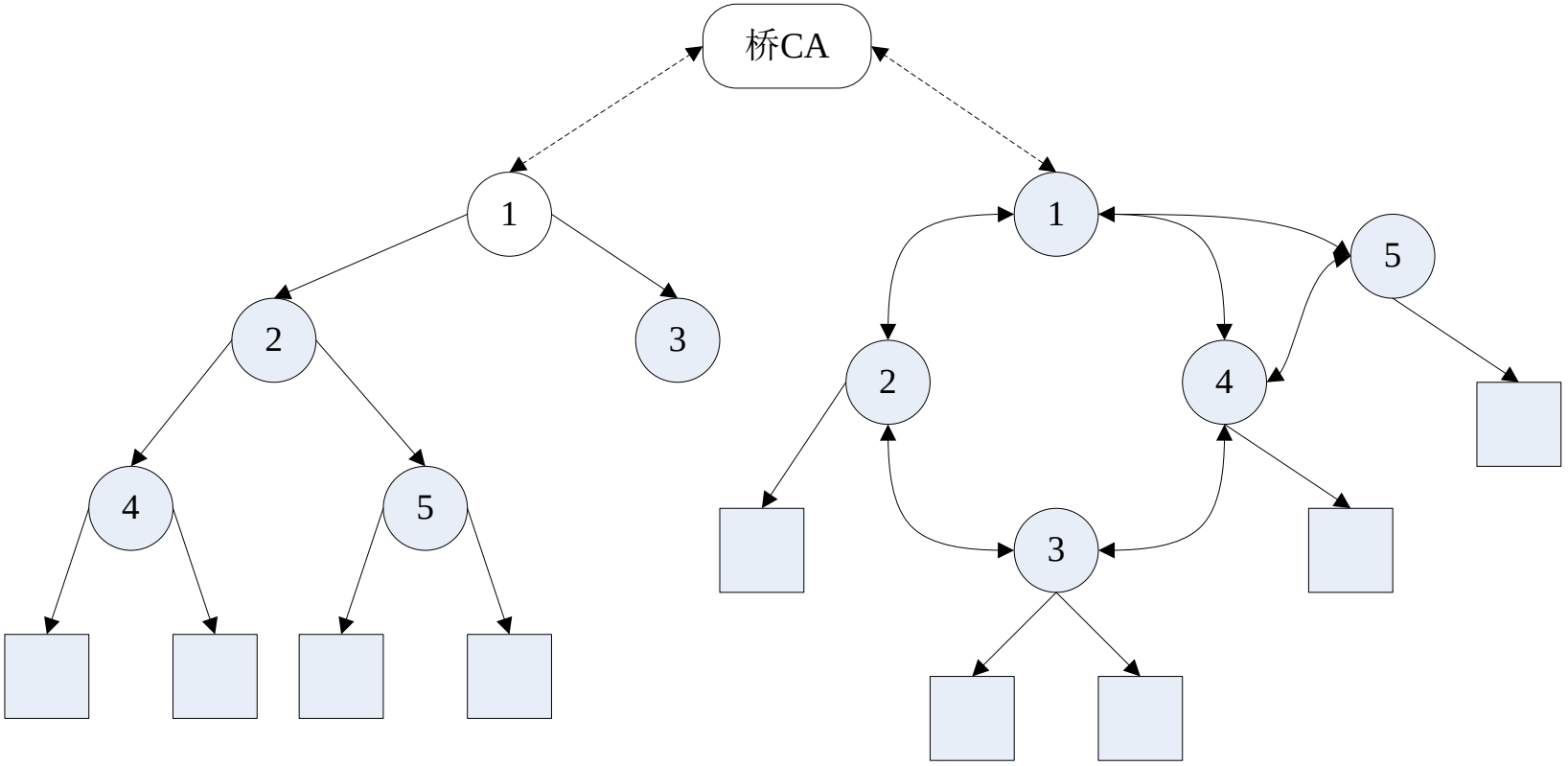


图3-14 桥模型

5.信任链模型

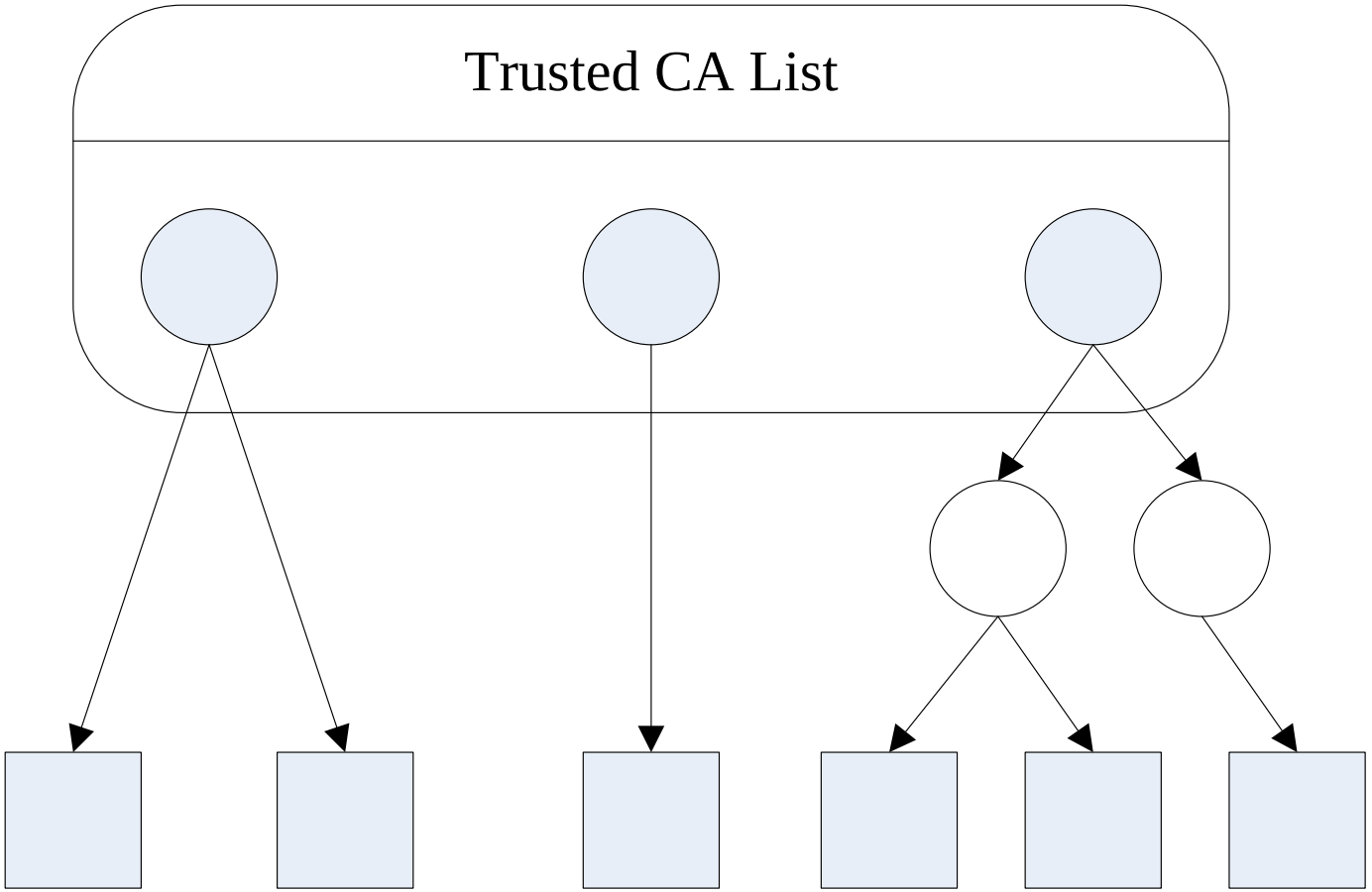
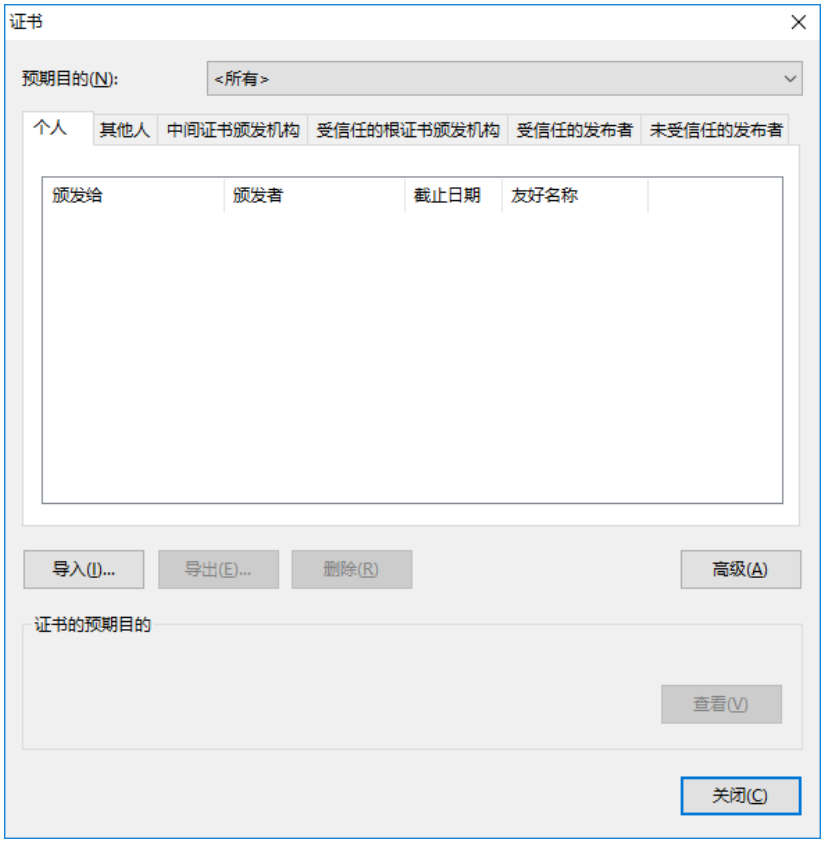
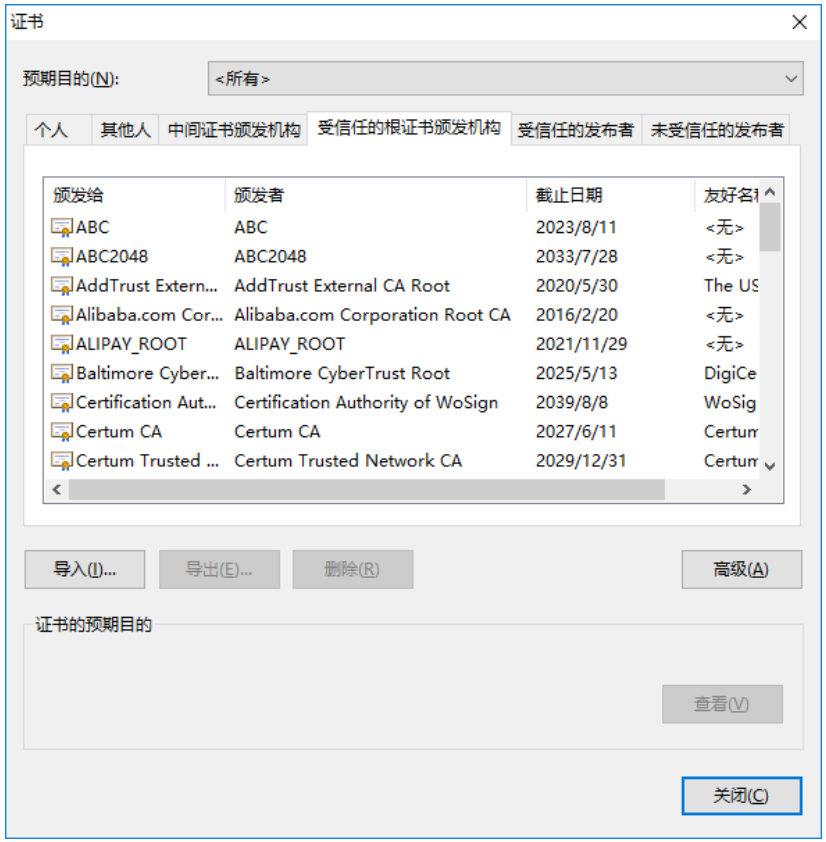


图3-15 信任链模型

Chrome浏览器中的证书

Chrome浏览器中的证书





思考：如何保证根证书可信？

第3章 作业

- 作业

  1.计算机系统对人进行认证的主要方法有哪些？

  5.什么是证书？证书的基本功能是什么？

  6.简述X.509证书包含的信息。

  9.简述主要的PKI信任模型。

- 实践（自己研究，不考核）
  - 查看浏览器中的证书。