# 第4章 身份认证

罗文坚

# 主要内容

- **4.1 概述**
- **4.2 认证协议**
  - **4.2.1 基于对称密钥的认证协议**
  - **4.2.2 基于公开密钥的认证协议**
- **4.3 公钥基础设施PKI**
  - **4.3.1 PKI体系结构**
  - **4.3.2 基于X.509的PKI系统**

# 概述

- 问题的提出：什么是身份认证？
- 身份认证是证实用户的真实身份与其所声称的身份是否相符的过程。

- 身份认证的依据应包含只有该用户所特有的、并可以验证的特定信息。
  - 用户所知道的或所掌握的信息（**Something the user know**），如密码、口令等；
  - 用户所拥有的特定东西（**Something the user possesses**），如身份证、护照、密钥盘等；
  - 用户所具有的个人特征（**Something the user is or How he behaves**），如指纹、笔迹、声纹、虹膜、**DNA**等。

# 概述

- 目前身份认证技术主要包括三类：
  - 基于口令的认证技术： 简单灵活，但容易泄露。
  - 基于密码学的认证技术：包括基于对称密钥的认证、基于公开密钥的认证协议。
  - 生物特征的认证技术：依附于人体，不易伪造，不易模仿！

# 身份认证的分类

- 根据认证条件的数目分类：
  - 仅通过一个条件的相符合来证明一个人的身份，称之为单因子认证；
  - 通过两种不同条件来证明一个人的身份，称之为双因子认证；
  - 通过组合多种不同条件来证明一个人的身份，称之为多因子认证。
- 根据认证数据的状态来看：
  - 静态数据认证：指用于识别用户身份的认证数据事先已产生并保存在特定的存储介质上；
  - 动态数据认证：指用于识别用户身份的认证数据不断动态变化，每次认证使用不同的认证数据，即动态密码。动态密码由一种称为动态令牌的专用设备（硬件或软件）产生，其产生动态密码的算法与认证服务器采用的算法相同。

# 主要内容

- 4.1 概述
- 4.2 认证协议
  - 4.2.1 基于对称密钥的认证协议
  - 4.2.2 基于公开密钥的认证协议
- 4.3 公钥基础设施PKI
  - 4.3.1 PKI体系结构
  - 4.3.2 基于X.509的PKI系统

# 认证协议

- 以网络为背景的认证技术的核心基础是密码学。
  - 对称密码和公开密码是实现用户身份识别的主要技术。

- 实现认证必须要求示证方和验证方遵循一个特定的规则来实施认证，这个规则被称为认证协议。

- 认证过程的安全取决于认证协议的完整性和健壮性。

# 基于对称密钥的认证协议

- 示证方和验证方共享密钥，通过共享密钥来维系彼此的信任关系，实际上认证就是建立某种信任关系的过程。

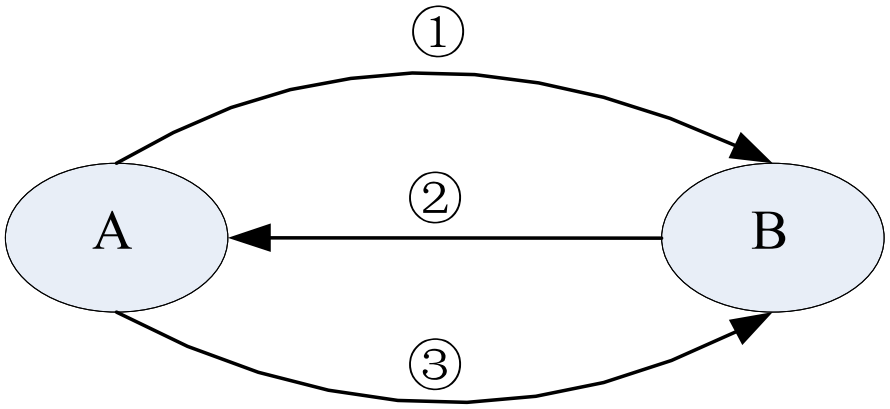
- 在只有少量用户的封闭式网络系统中，各用户之间的双人共享密钥的数量有限，可以采用挑战-应答方式来实现认证；

- 对于规模较大的网络系统，一般采用密钥服务器的方式来实现认证，即依靠可信的第三方完成认证。

- 有关符号表示：
  - $\mathbf{A} \rightarrow \mathbf{B}$：表示A向B发送一条信息。
  - $E_k(x)$：使用共享秘钥$k$对信息串$x$加密。
  - $x \| y$：表示$x$和$y$相连接。

# 基于挑战-应答方式的认证协议

1. 由验证方生成一个大的随机数据串，即挑战，将挑战发送给示证方。
2. 示证方使用共享秘钥加密挑战，然后回送给验证方，
3. 验证方通过解密密文得到挑战，通过验证挑战的正确与否，来认证示证方的身份。




① A ⟶ B：IDa ‖ IDb

② B ⟶ A：Nb

③ A ⟶ B：$E_k$(Nb)

说明：$ID_a$和$ID_b$分别是A和B的网络用户标识。

# Needham-Schroeder认证协议

- 所有的使用者共同信任一个公正的第三方，此第三方被称为认证服务。
- 每个使用者需要在认证服务器**AS**（**Authentication Server**）上完成注册，**AS**保存每一个用户的信息并与每一个用户共享一个对称密钥。
- 用户和**AS**之间的信任关系依靠它们的共享秘钥来维系。
- 有关符号：
  - **KDC**（**Key Distribution Center**）为**AS**的秘钥分配中心，主要功能是为用户生成、分发通信秘钥。
  - $\mathbf{ID_A}$和$\mathbf{ID_B}$分别是**A**和**B**的网络用户标识。

# Needham-Schroeder协议描述

① A → KDC： $ID_A \| ID_B \| N_1$ ；

– A通知KDC要与B进行安全通信，$N_1$为临时值。

② KDC → A： $E_{K_a}\left[K_s \| ID_B \| N_1 \| E_{K_b}[K_s \| ID_A]\right]$;

– $K_s$为会话秘钥，$N_1$表示当次申请，用$K_a$加密保证安全性；

– $K_b$加密转发给B的内容，该内容只能由B和AS还原。

③ A → B： $E_{K_b}[K_s \| ID_A]$； A转发KDC给B的内容。

④ B → A： $E_{K_s}[N_2]$;

– B用$K_s$加密挑战值$N_2$，发给A并等待A的回应认证信息。

⑤ A → B： $E_{K_s}[f(N_2)]$ ;

– A还原$N_2$后，根据事先的约定f(x)，比如f(x)=x-1，计算$f(N_2)$，使用$K_s$加密后，回应B的挑战，完成认证，随后A和B使用$K_s$进行加密通讯。

# Needham-Schroeder协议的漏洞

- Needham-Schroeder协议存在漏洞。

- 假定攻击方C掌握了A和B之间通信的一个老的会话。
  - C可以在第3步冒充A利用老的会话欺骗B。
  - 除非B记住所有以前使用的与A通信的会话密钥，否则B无法判断这是一个重放攻击。

# Kerberos

- Kerberos的设计目标是用对称密钥系统为客户机/服务器应用程序提供强大的第三方认证服务。
  - 每个用户或应用服务器与Kerberos分享一个对称密钥。
  - Kerberos由两个部分组成：
    - 认证服务器AS（Authentication Server）
    - 票据授予服务器TGS（Ticket Granting Server）。
  - 允许一个用户通过交换加密消息，在整个网络上与另一个用户或应用服务器互相证明身份，Kerberos给通讯双方提供对称密钥。
  - 票据Ticket是客户端访问服务器时，提交的用于证明自己身份，并可传递通信会话秘钥的认证资料。
    - AS负责签发访问TGS服务器的票据，TGS负责签发访问其它应用服务器的票据。

# 协议内容

- 第一阶段 身份验证服务交换：完成身份认证，获得访问TGS的票据。

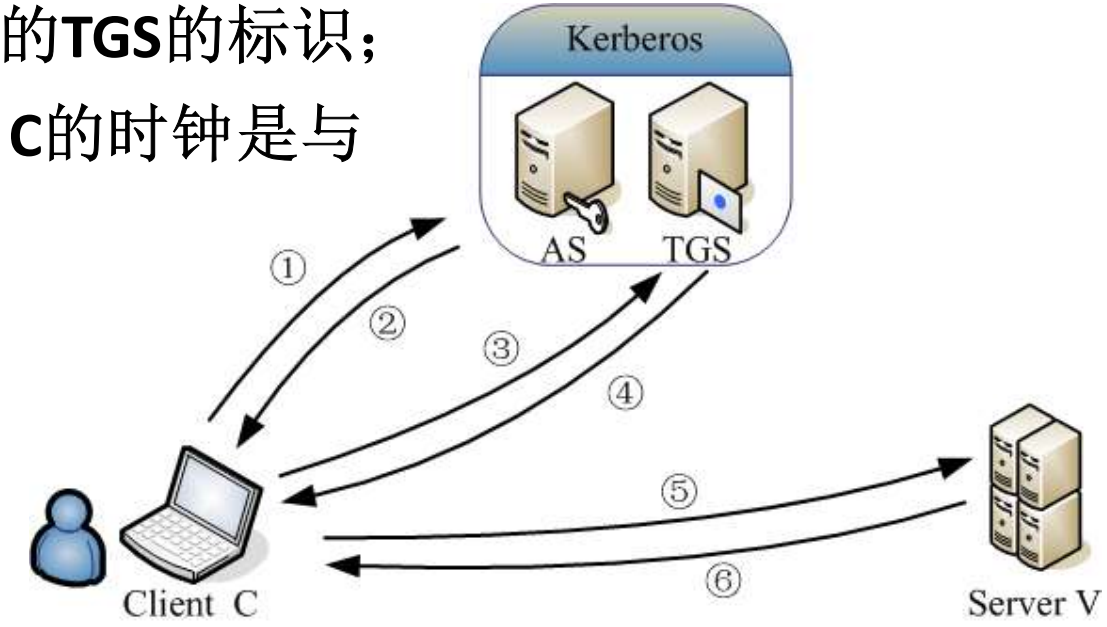
1. $\mathbf{C \rightarrow AS}$：$\mathbf{ID_C \parallel ID_{tgs} \parallel TS_1}$
2. $\mathbf{AS \rightarrow C}$：$\mathbf{E_{K_C}[K_{c,tgs} \parallel ID_{tgs} \parallel TS_2 \parallel Lifetime_2 \parallel Ticket_{tgs}]}$

➢ 步骤1为请求TGS票据。

– $\mathbf{ID_C}$：Client C的用户标识；

– $\mathbf{ID_{tgs}}$：用户请求访问的TGS的标识；

– $\mathbf{TS_1}$：让AS验证Client C的时钟是与AS的时钟是否同步。



# 协议内容

- 第一阶段 身份验证服务交换
  1. $\mathbf{C \rightarrow AS:\ ID_C \parallel ID_{tgs} \parallel TS_1}$
  2. $\mathbf{AS \rightarrow C:\ E_{K_C}[K_{c,tgs} \parallel ID_{tgs} \parallel TS_2 \parallel Lifetime_2 \parallel Ticket_{tgs}]}$

  ➢ 步骤2为返回TGS票据。

  – $\mathbf{K_{c,tgs}}$：由AS产生，用于在TGS和Client C之间安全交换信息。

  – $\mathbf{ID_{tgs}}$：确认这个ticket是为特定TGS制作的。

  – $\mathbf{TS_2}$：告诉用户该ticket签发的时间。

  – $\mathbf{Lifetime_2}$：告诉用户该ticket的有效期。

  – $\mathbf{Ticket_{tgs}}$：用户用来访问TGS的ticket，可重用，避免多次认证输入口令，且$\mathbf{Ticket_{tgs} = E_{K_{tgs}}[K_{c,tgs} \parallel ID_C \parallel AD_C \parallel ID_{tgs} \parallel TS_2 \parallel Lifetime_2]}$，其中$\mathbf{ID_C}$指明该ticket的真正主人，$\mathbf{AD_C}$为Client C的网络地址。

# 协议内容

- 第二阶段 票据授予服务交换：获得访问应用服务器的票据。

  3. $C \rightarrow TGS$： $ID_V \parallel Ticket_{tgs} \parallel Authenticator_c$

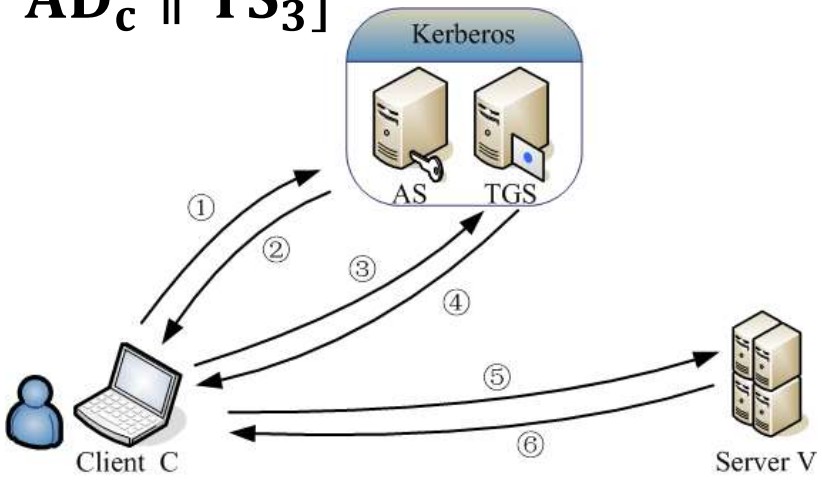
  4. $TGS \rightarrow C$： $E_{K_{c,tgs}}[K_{c,v} \parallel ID_v \parallel TS_4 \parallel Ticket_v]$

  ➢ 步骤3为请求应用服务器票据。

  – $ID_v$：告诉TGS用户要访问应用服务器V。

  – $Ticket_{tgs}$：向TGS证实该用户已被AS认证；

  – $Authenticator_c$：由用户生成，用于验证时效性，且 $Authenticator_c = E_{K_c,tgs}[ID_C \parallel AD_c \parallel TS_3]$



# 协议内容

- 第二阶段 票据授予服务交换：获得访问应用服务器的票据。

  3. $\mathbf{C \rightarrow TGS}$：$\mathbf{ID_V \parallel Ticket_{tgs} \parallel Authenticator_c}$

  4. $\mathbf{TGS \rightarrow C}$：$E_{K_{c,tgs}}[\mathbf{K_{c,v} \parallel ID_v \parallel TS_4 \parallel Ticket_v}]$

  ➢ 步骤4为返回应用服务器票据。

  – $\mathbf{K_{c,v}}$：由TGS生成，用于Client C和Server V之间安全交换信息。

  – $\mathbf{ID_v}$：确认该ticket是签发给server V的。

  – $\mathbf{TS_4}$：告诉用户该ticket签发的时间。

  – $\mathbf{Ticket_v}$：用户用以访问应用服务器V的ticket，且

  $\mathbf{Ticket_v} = \mathbf{E_{K_v}}[K_{c,v} \parallel \mathbf{ID_C} \parallel \mathbf{AD_C} \parallel ID_v \parallel TS_4 \parallel \mathbf{Lifetime_4}]$，

  - $\boldsymbol{E_{K_v}}[\,]$：Ticket用只有TGS和Server V共享的秘钥加密，以防篡改。

# 协议内容

- 第三阶段 客户与服务器身份验证交换：获得服务。

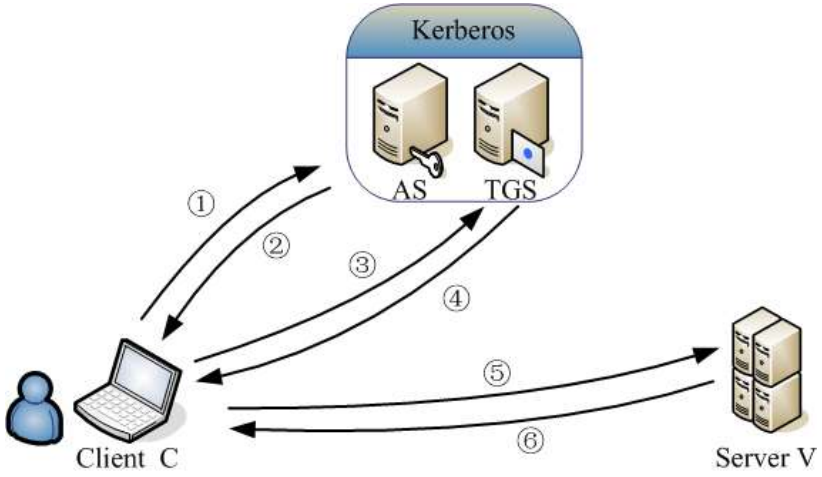
  5. $\mathbf{C \rightarrow V}$： $\mathbf{Ticket_v \parallel Authenticator_c}$

  6. $\mathbf{V \rightarrow C}$： $E_{K_{c,v}}[TS_5 + 1]$ **( for mutual authentication)**

  ➢ 步骤**5**为向应用服务器发起服务请求。

  – $\mathbf{Ticket_v}$：向服务器证实该用户已被**AS**认证。

  – $\mathbf{Authenticator_c}$：由**Client C**生成用于验证时效性，且 $\mathbf{Authenticator_c} = E_{K_{c,v}}[ID_C \parallel AD_C \parallel TS_5]$，其中$E_{K_{c,v}}[\,]$用**Client C**和**Server V**的共享秘钥加密，以验证身份和保护本信息。



# 协议内容

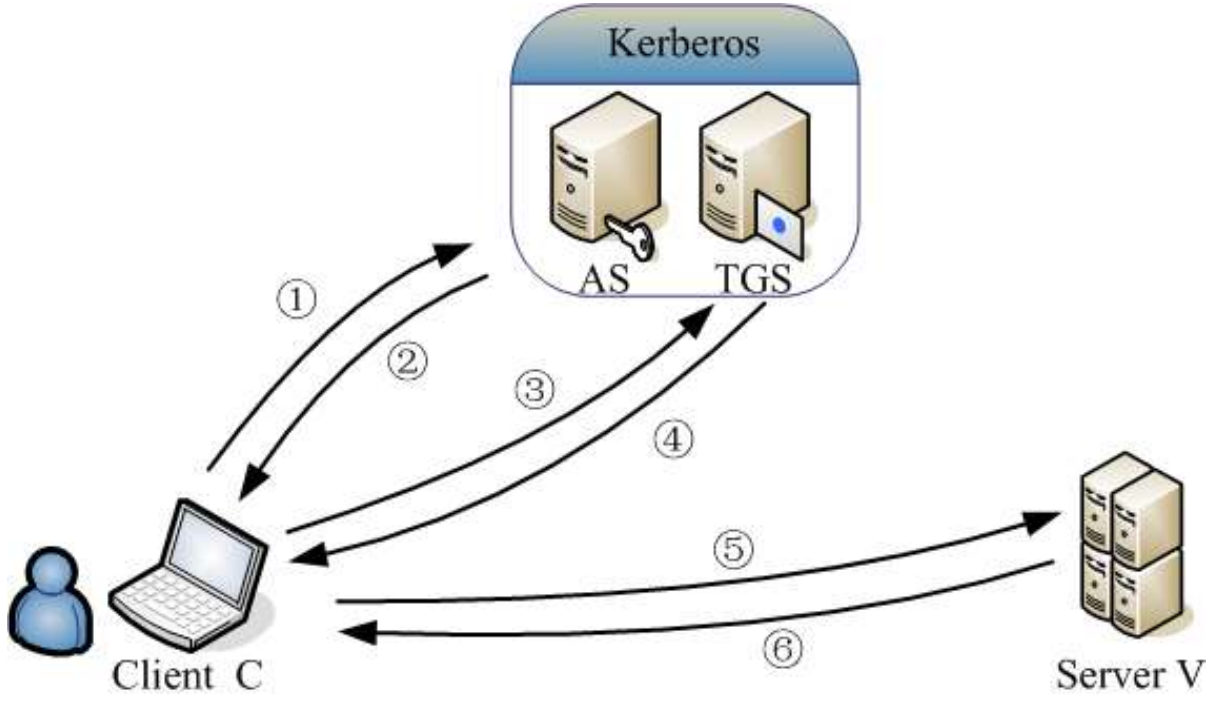
- 第三阶段 客户与服务器身份验证交换：获得服务。

  5. $\mathbf{C \rightarrow V}$： $\mathbf{Ticket_v \parallel Authenticator_c}$

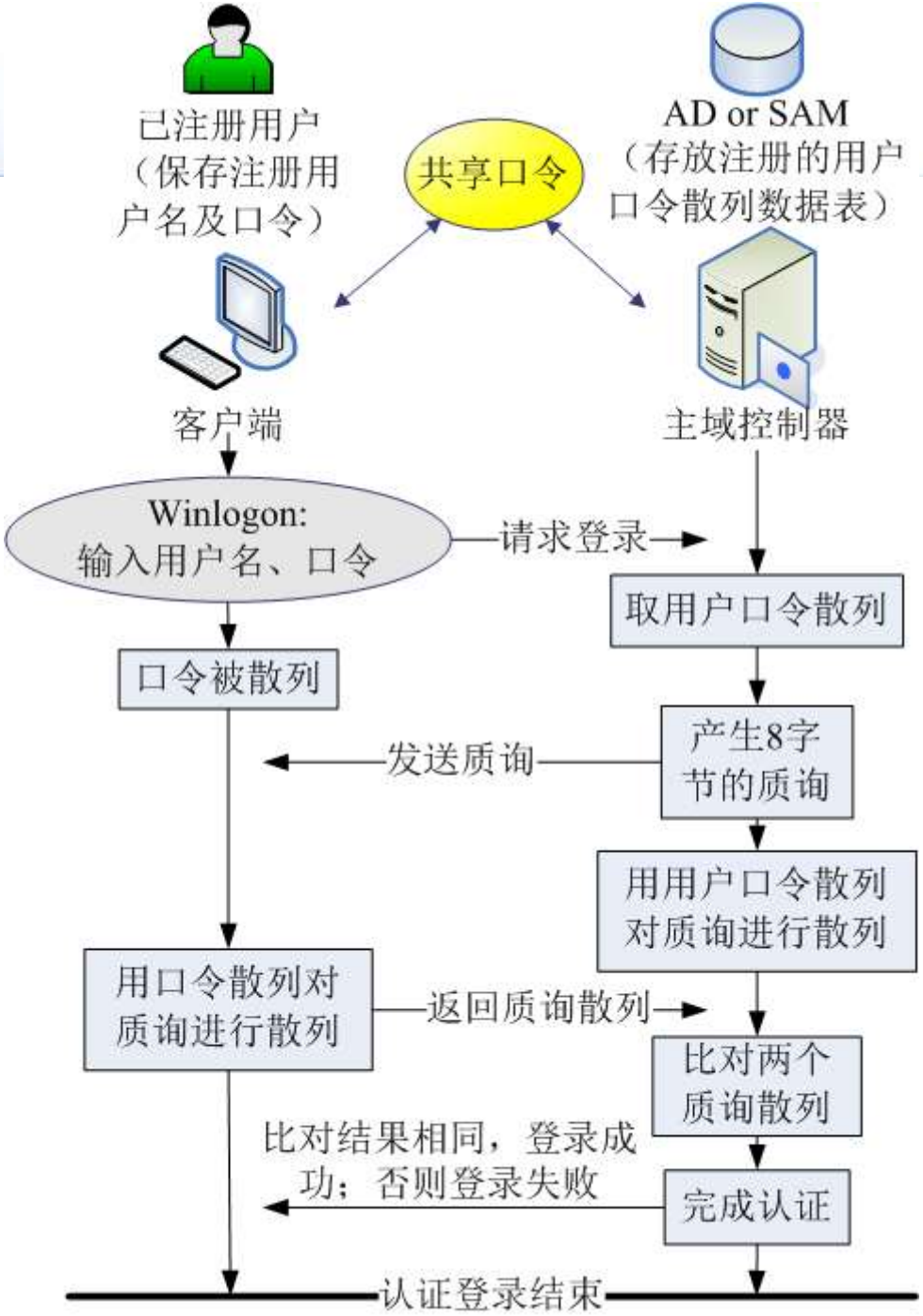
  6. $\mathbf{V \rightarrow C}$： $E_{K_{c,v}}[\mathbf{TS_5 + 1}]$ **( for mutual authentication)**

  ➢ 步骤**6**为服务器对客户机可选的身份验证。

  – $\mathbf{TS_5 + 1}$： 向**Client C**证明这不是重放攻击的应答。



# Windows用户登录认证过程




（也称加密）

# 基于公开密钥的认证协议

- 基于公开密钥体制下的认证协议通常有两种认证方式：

  - 方式一：实体A需要认证实体B，A发送一个明文挑战消息（也称挑战因子，通常是随机数）给B，B接收到挑战后，用自己的私钥对挑战明文消息加密，称为签名；B将签名信息发送给A，A使用B的公钥来解密签名消息，称为验证签名，以此来确定B是否具有合法身份。

  - 方式二：实体A将挑战因子用实体B的公钥加密后发送给B，B收到后是用自己的私钥解密还原出挑战因子，并将挑战因子明文发还给A，A可以根据挑战因子内容的真伪来核实B的身份。

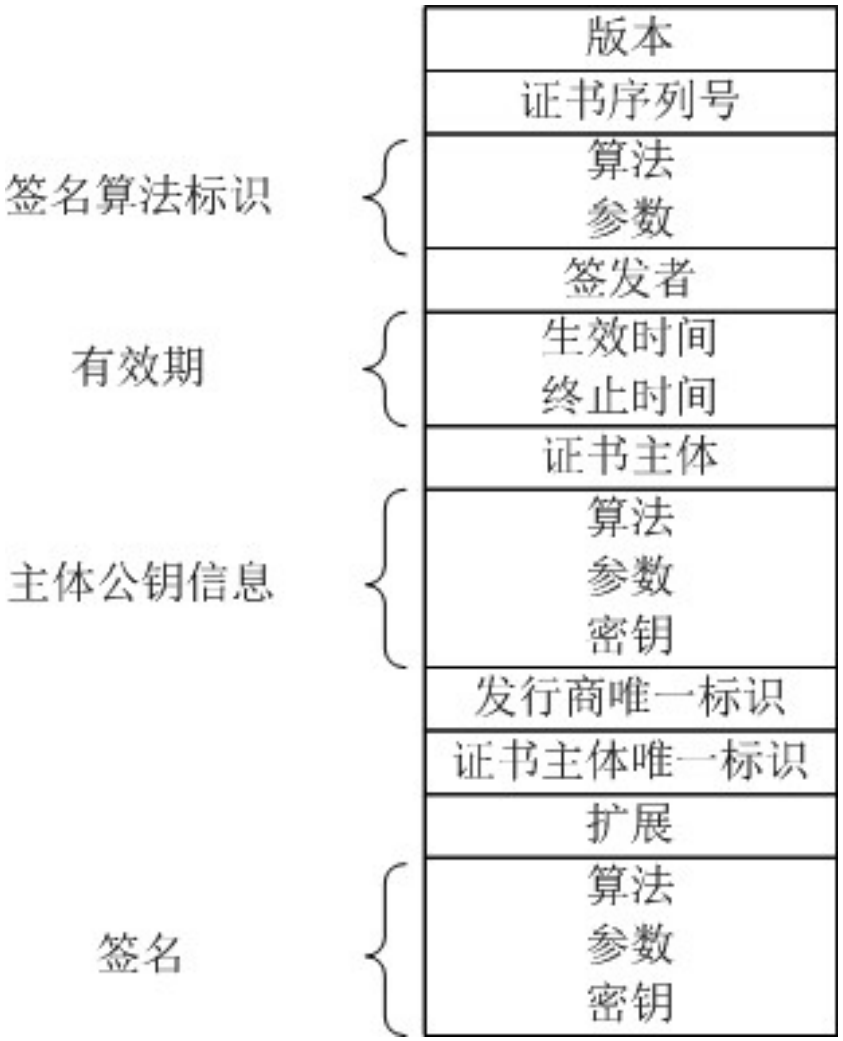
# Needham-Schroeder公钥认证

- ① A → B： $E_{KUb}[ID_a \| R_a]$；
  - A使用B的公钥加密A的标识$ID_a$和挑战$R_a$，确保只有B才能使用私钥解密。
- ② B → A： $E_{KUa}[R_a \| R_b]$；
  - B使用A的公钥加密A的挑战$R_a$和B的挑战$R_b$，发送给A，确保只有A才能使用其私钥解密。
- ③ A → B： $E_{KUb}[R_b]$；
  - A还原出$R_b$后，再使用B的公钥加密$R_b$，作为验证应答信息发送给B。

# 基于CA数字证书的认证协议

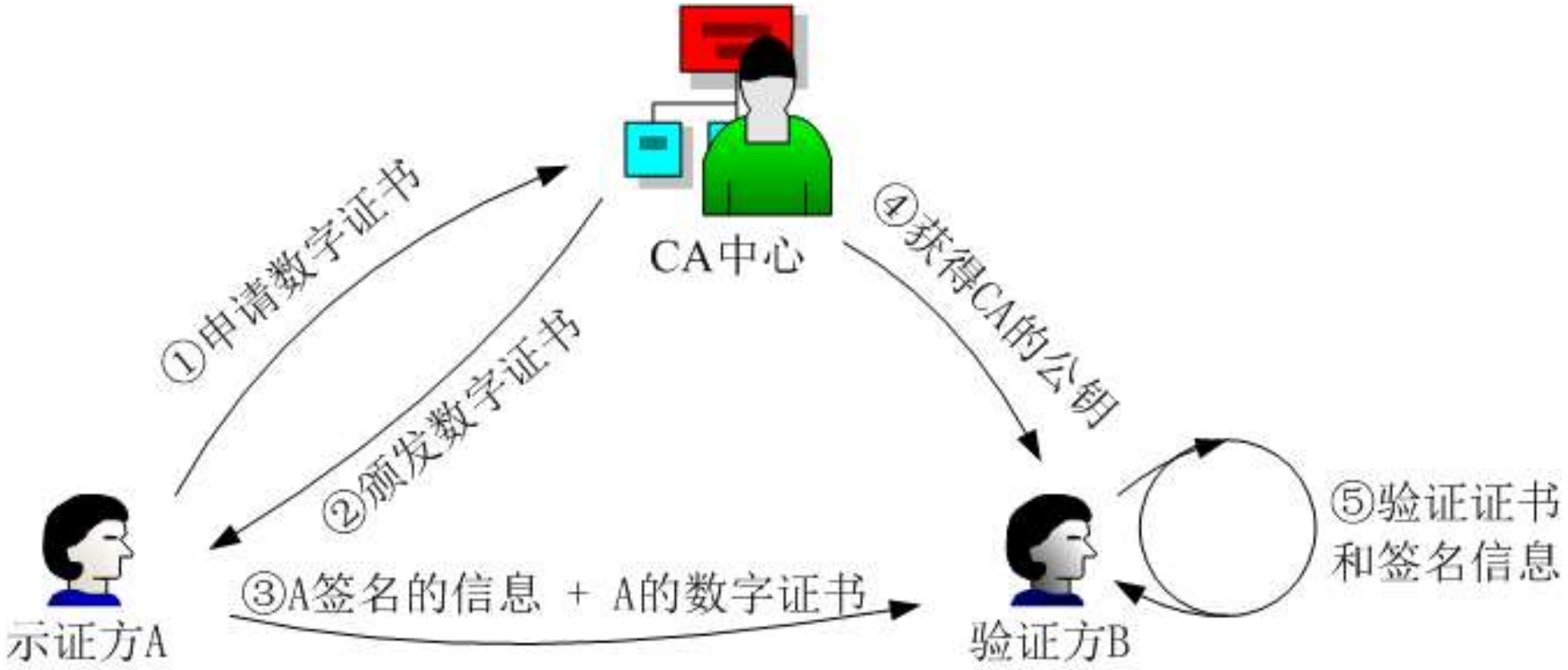
- 基于CA数字证书的认证协议属于公开秘钥的认证协议范畴，只是引入了一个可信的第三方来管理公钥并提供仲裁。在实际的网络环境中，公钥是采用数字证书（Certificate）的形式来完成发布的。

- 数字证书是一个经过权威的、可信赖的、公正的第三方机构（即CA认证中心，Certificate Authority）签名的包含拥有者信息及公开密钥的文件。




X. 509 V3证书格式

# 基于数字证书进行身份认证的过程




- 通过5个环节，B可以确认A的身份及其签名的信息。

# 基于数字证书进行身份认证的过程

1. A提交资料，申请证书。
2. CA审核A的资料，颁发用CA私钥签过名的数字证书。
   - 该数字证书包含了A的身份信息和A的公钥。由于使用了CA的私钥签名，因此其他人无法伪造。
3. A使用私钥对特定信息进行签名，连同数字证书一起发送给B，B为验证方。
4. B为了能够核实A的数字证书的真伪，必须先获得CA的公钥。
5. B使用CA的公钥对A的数字证书进行合法性验证，通过后获得A的公钥，对A签过名的特定信息进行认证，进而确认A的身份及其签名的信息。

# 主要内容

- 4.1 概述
- 4.2 认证协议
  - 4.2.1 基于对称密钥的认证协议
  - 4.2.2 基于公开密钥的认证协议
- 4.3 公钥基础设施PKI
  - 4.3.1 PKI体系结构
  - 4.3.2 基于X.509的PKI系统

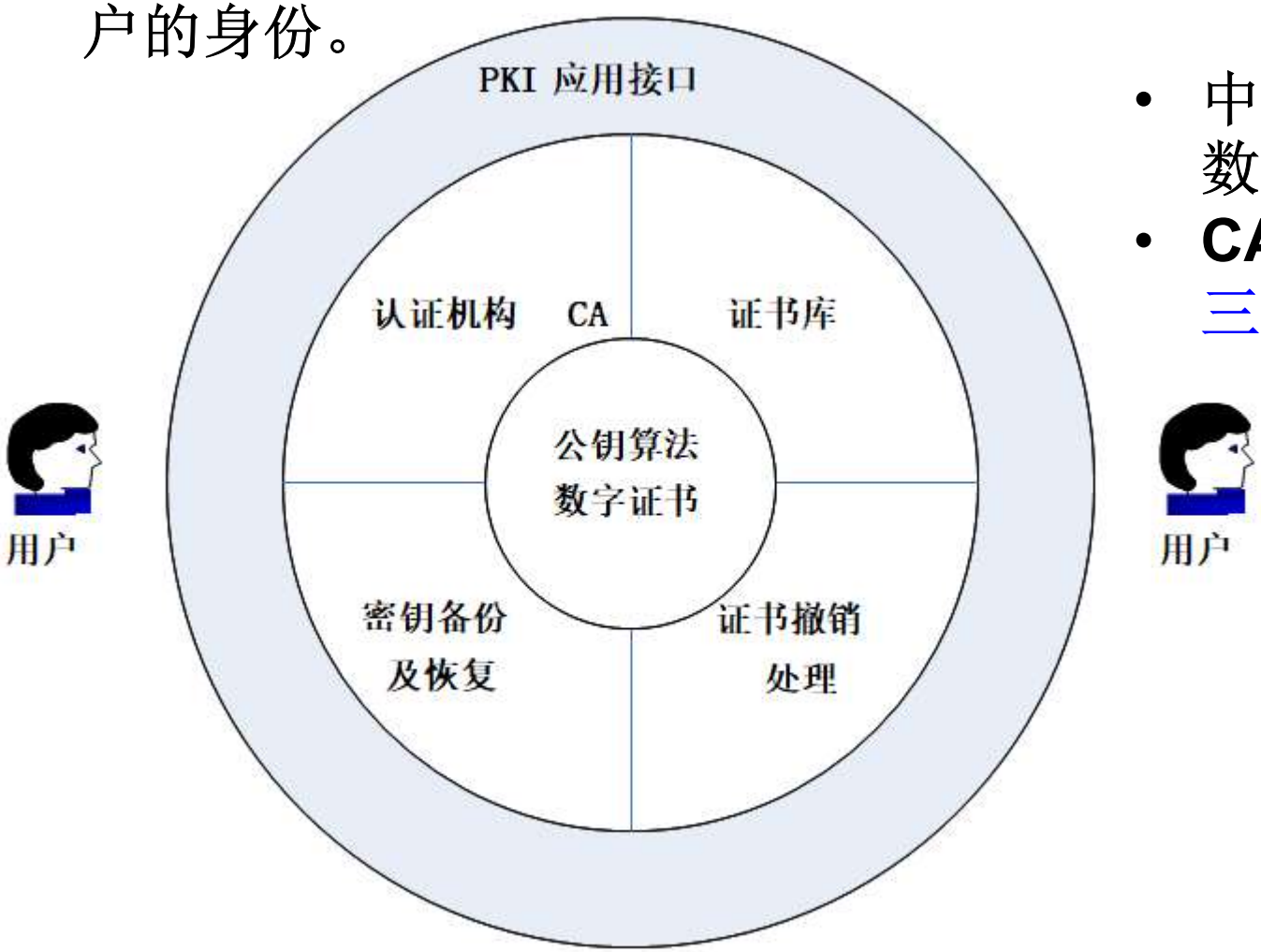
# 公钥基础设施PKI

- 为了解决Internet上电子商务等应用的安全问题，世界各国经过多年的研究，初步形成了一套完整的Internet安全解决方案，即目前被广泛采用的公钥基础设施（Public Key Infrastructure，PKI）。

- PKI是一种遵循一定标准的密钥管理基础平台，为所有网络应用提供加密和数字签名等密码服务所必需的密钥和证书管理。
  - PKI就是利用公钥理论和技术建立的提供安全服务的基础设施。
  - 用户可利用PKI平台提供的服务进行安全的电子交易、通信和互联网上的各种活动。

# PKI体系结构

- PKI采用数字证书技术来管理公钥，通过CA认证中心把用户的公钥和用户的其他标识信息捆绑在一起，在互联网上验证用户的身份。




- 中心是公钥算法和数字证书技术。
- CA认证中心是第三方的可信任机构。

# PKI体系结构

1. 认证机构CA是PKI的核心执行机构，也称为认证中心，其主要功能包括数字证书的申请注册、证书签发和管理。
   - 其工作内容包括验证并标识证书申请者的身份，对证书申请者的信用度、申请证书的目的、身份的真实可靠性等问题进行审查，确保证书与身份绑定的正确性。
   - 当服务范围较大时，CA可以拆分出证书申请注册机构（Registration Authority，RA），专门负责证书的注册申请和撤销申请等管理工作。
2. 证书库（Repository）是CA颁发证书和撤销证书的集中存放地，是网上的公共信息库，可供公众进行开放式查询。
   - 查询的目的有两个：一是想得到与之通信实体的公钥，二是要验证通信对方的证书是否已进入黑名单。
   - 证书库一般采用LDAP（Lightweight Directory Access Protocol）协议搭建分布式的目录系统。

# PKI体系结构

3. 秘钥备份及恢复：当用户证书生成时，秘钥被CA备份存储。当秘钥丢失需要恢复时，用户只需向CA提出申请，CA就会为用户自动进行秘钥恢复。

4. 证书撤销处理：证书和秘钥都有一定的生成期限。当用户的私钥泄露或公司职员离职时，都需要撤销原CA证书。被撤销的CA证书进入证书库的黑名单，公众可查询核实。

5. PKI应用接口是使用者和PKI交互的唯一途径，可以看成是PKI的客户端软件。使用者在其计算机上安装PKI的客户端软件。

- PKI平台包括以上四个基本功能模块和一个应用接口模块。

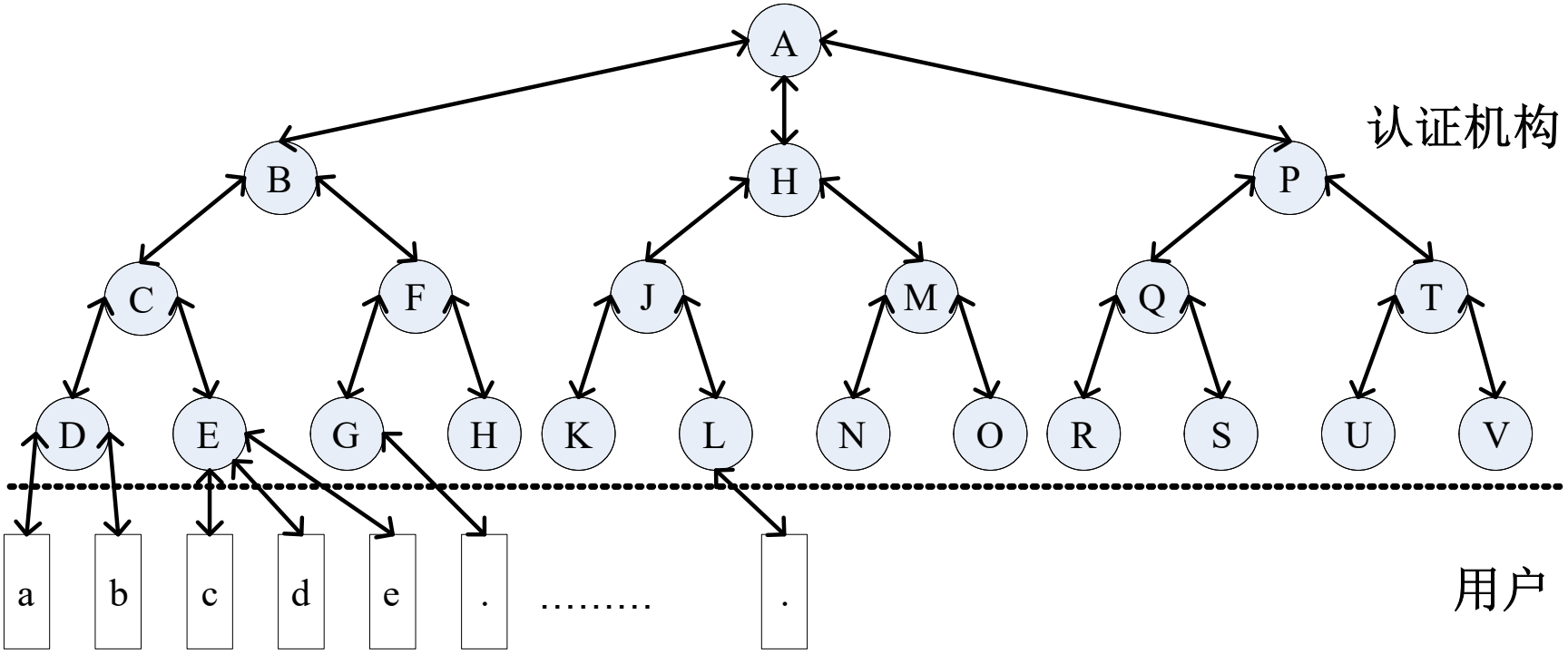
# 基于X.509的PKI系统

- X.509是国际电信联盟-电信（ITU-T）部分标准和国际标准化组织（ISO）的证书格式标准。

- X.509的主要作用是确定了公钥证书结构的基准。
  - 当前使用的版本是X.509 V3。
  - X.509 V3证书包括一组按预定义顺序排列的强制字段，还有可选扩展字段。即使在强制字段中，X.509证书也具有很大的灵活性，因为它为大多数字段提供了多种编码方案。

# X.509的CA目录的层次结构

基于X. 509的层次型认证机构分布




- 用户a的证书链可以使用下面的形式表达：

  $KR_A《CA_B》KR_B《CA_C》KR_C《CA_D》KR_D《CA_a》$

  – $KR_X$表示X的私钥签名，$CA_X$表示X的证书。

# X.509的CA目录的层次结构

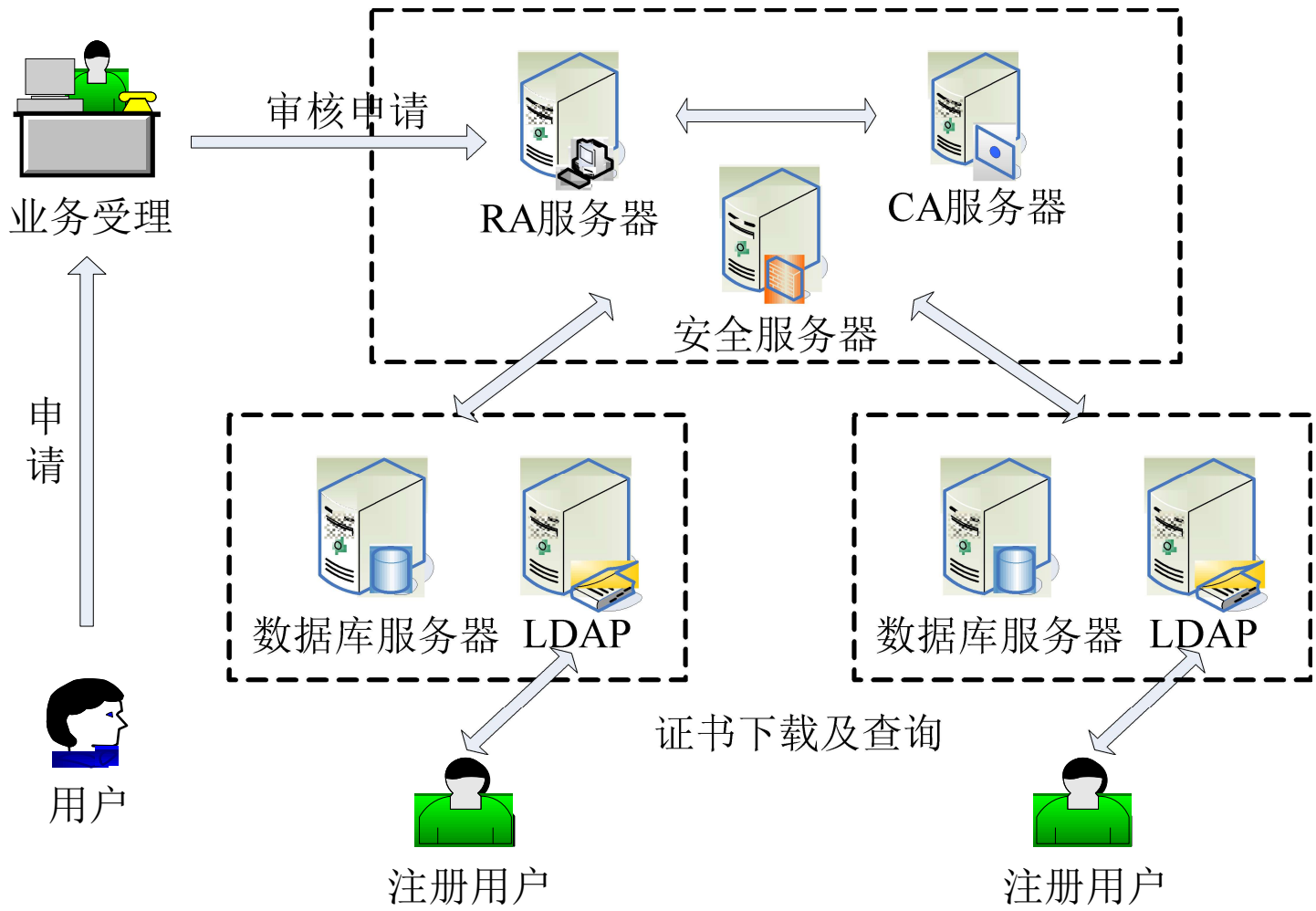
- 如果某用户x希望验证用户a的证书。

1. 用户a的CA证书是认证机构D签发的，用户x只要得到认证机构D的公钥，就可以验证用户a的证书中D的签名，即完成对用户a的证书的认证，从而得到用户a的公钥。

2. 假如用户x不能确定D的公钥，就必须查看D的证书。
   - 由于D的证书是由认证机构C签发的，因此，用户x需要使用C的公钥验证D的证书并得到其公钥。

3. 以此类推，最坏的情况是用户x需要使用认证机构A的公钥，而A是此认证机构的根，A的证书也叫根证书，是使用其私钥自签名产生的。
   - 用户在使用CA证书之前必须先下载安装A的证书，同时系统会自动加载保存认证机构A的公钥。

# 一个典型的PKI模型


审核申请
业务受理
RA服务器
CA服务器
安全服务器
申请
数据库服务器
LDAP
数据库服务器
LDAP
证书下载及查询
用户
注册用户
注册用户

# PKI系统功能

1. 接收验证用户数字证书的申请；
2. 确定是否接受用户数字证书的申请；
3. 向申请者颁发（或拒绝颁发）数字证书；
4. 接收、处理用户的数字证书更新请求；
5. 接收用户数字证书的查询、撤销；
6. 产生和发布证书的有效期；
7. 数字证书的归档；
8. 密钥归档；
9. 历史数据归档。

# 作业

1. “挑战应答方式”认证和**Needham-Schroeder**对称密钥认证协议的区别？模仿**Needham-Schroeder**设计一个三方通讯认证协议。
2. 解释详细**Kerberos**认证协议过程。
3. 什么是数字证书，如何使用数字证书进行身份认证？
4. 什么是**PKI**，它包含那些主要功能，如何工作？
5. 什么是证书链，**X.509**是如何实现证书认证的？