第 3 章　网络身份认证

进入网络系统的用户首先需要进行身份认证，获取进入网络大门的许可。常见的网络身份认证方式有口令认证、IC 卡认证、基于生物特征的认证和双因子认证等。网络环境下的身份认证一般通过某种身份认证协议来实现。身份认证协议一般基于密码相关技术实现，定义了参与认证服务的各通信方在身份认证过程中需要交换的所有消息的格式、这些消息发生的次序以及消息的语义。本章最后以单点登录为例说明网络身份认证的应用。

本章主要内容：

- 网络身份认证概述
- 常用网络身份认证技术
- 网络身份认证协议
- 单点登录

3.1　网络身份认证概述

3.1.1　身份认证案例

随着互联网用户对于网络的使用程度在不断加深，国内各行业也积极拥抱互联网，将互联网的技术和思维运用到生产、运输、营销、服务等环节，新技术、新模式、新业态正在不断涌现。互联网在给人们带来便利的同时，也日益威胁到个人和企业的信息安全。全球互联的网络世界中也充斥着计算机病毒和黑客，个人信息泄露、非法窃听和电子欺诈等案例时有发生。

2011 年，360 安全卫士官方微博发布了一条紧急通知，称 CSDN（Chinese Software Developer Network）网站历史数据库遭黑客入侵，600 余万用户数据泄密。其主要原因是：在 2009 年 4 月之前，CSDN 网站以明文方式存储和传输用户的个人信息，未采取任何加密措施，黑客入侵系统后，可以轻易地获取用户信息，并公开用户信息数据库。以同样原因泄露用户信息的类似事件在近几年频频发生，2012 年近 2 亿 UC 手机浏览器用户面临泄密威胁，主要原因在于 UC 浏览器的快捷上网功能采用了一种压缩中转技术，当用户通过 UC 浏览器登录 Gmail 等网站时，UC 浏览器会把用户访问的 URL 地址和提交的信息发送到附近的一台 UC 服务器，这里存在的漏洞是，UC 浏览器手机端和 UC 服务器之间包括用户名和密码在内的所有信息均为明文传输，这使得 UC 浏览器和 UC 服务器之间的通信可以被监听和抓包，第三方可以通过这种方法获取手机用户的账户、密码等敏感信息，用户通过 UC 浏览器登录的任何网站都会被监听，包括邮箱、网站后台、网银、网上支付等。

除此之外，由于用户的密码强度过低而导致信息泄露的案例也屡见不鲜。例如，2013 年，Adobe 由于密码设定强度过低而遭到黑客大量破解，可任意获取用户的相关数据，约有 1.52 亿用户受到影响。而以系统内部人员身份进行非法攻击和信息倒卖的事件也时有发

生，2014 年，eBay 部分员工的登录凭证遭黑，导致内部数据库可被截取，攻击时间长达两个月，约有 1.45 亿用户受到影响。2014 年，支付宝前技术人员李某利用其工作之便，多次在公司后台下载支付宝用户的资料，资料内容超过 20GB，随后将用户信息多次出售给电商公司、数据公司。

由于各种原因而引发的用户身份信息泄露不仅危及人们的人身安全和财产安全，甚至能够危及国家的安全和发展。而且身份认证作为保护用户信息安全和系统安全的第一道防线，在网络安全建设过程中占据重要地位。因此，对网络身份认证的进一步研究已经迫在眉睫。

网络环境下的身份认证就是指通过一定的认证技术来确认相关用户和通信实体身份，进而确定该用户和通信实体是否具有对某种资源的访问和使用权限的过程。现实生活中，每个人都拥有独一无二的身份，对人的身份认证最常见的形式是查验各种证件实物（如身份证、工作证等）。而在计算机网络环境中，用户和网络设备的身份信息都是由一组特定的数据表示的数字标识，对他们的身份认证就是对其数字身份的验证，即验证他们的现实身份与数字身份是否一致。身份认证技术就是用来解决如何保证用户和网络设备的数字身份与其现实身份相一致的方法。

身份认证其实包含两方面的内容，一是标识（identification），二是验证（authentication）。

（1）标识。用来代表实体的身份，就是要明确访问者是谁，系统中的实体标识必须具备唯一性和可辨认性特征。通过唯一标识符，系统可以识别出访问系统的每个用户或设备。例如，在网络环境中，网络管理员常用 IP 地址、网卡地址作为计算机用户的标识。

（2）验证。是系统对实体提供的标识（即身份）的真实性进行鉴别，以防止冒名顶替或恶意篡改。鉴别的依据是用户所拥有的特殊信息或实物，这些信息具有保密性，其他用户不能拥有。

3.1.2 身份认证的地位与作用

在计算机网络系统中，为了防止各种资源（如计算机硬件、软件、存储的数据等）未经授权而被泄露、使用、破坏，必须实现访问控制，使得只有经过授权的用户才能以被授权的方式进行访问。而访问控制的前提是能够识别用户的真实身份，然后系统才能根据不同的用户身份授予不同的访问权限，进而达到保护系统资源的目的。例如，通过 IP 地址的识别，网络管理员可以确定 Web 访问是内部用户访问还是外部用户访问。因此身份认证是有效实施其他安全策略（如建立安全信道、实施基于身份的访问控制和审计记录等）的前提和基础，是保护系统安全的第一道大门，在网络安全中占据十分重要的位置，它的失效可能导致整个系统的失败。

归纳起来，认证的主要用途有 3 个方面：

（1）验证用户身份，为网络系统访问控制服务提供支持。

（2）保证网络通信双方的真实性，防止假冒，为以后审计和责任追究提供支持。

（3）与其他安全机制相结合以保证数据的完整性和机密性，防止篡改、重放或延迟。

3.1.3 身份标识信息

计算机网络中的身份认证包括用户身份认证与设备身份认证。这里以用户身份认证为

例。认证过程就是通过与用户的交互获得标识用户身份的特殊信息(如用户名/口令组合、生物特征等),然后再对身份信息进行核对处理,根据处理结果确认用户身份是否正确。这里的正确指的是用户真实的身份与数字身份相对应。

常用的身份标识信息主要有 4 种:

- 用户知道的信息,如用户口令、PIN(Personal Identification Number,个人识别码)。
- 用户拥有的实物,一般是不可伪造的设备,如智能卡、磁卡等。
- 用户自身独一无二生的生物特征信息,如指纹、声音、视网膜等。
- 用户所处的位置,如 IP 地址(映射到一个特定子网)、MAC 地址(对应交换机上的特定端口)等。

上述每种标识信息都存在一些弱点,例如,口令容易泄露,实物会遗失,IP 地址可以被伪造,而基于生物特征信息进行认证的技术复杂且成本较高。在实际应用中,组合使用上述的前两种身份信息进行认证会显著提高安全性,通常称为双因子身份认证。例如,在 ATM 机上取款时,用户同时需要一个 PIN 号码和一个磁卡。即使有人获得了 PIN 号码,没有磁卡仍然不能访问。如果磁卡遗失或被偷,没有 PIN 号码也无法使用。当然,随着成本的降低,目前基于生物特征的认证也得到越来越广泛的应用,如基于指纹的识别、基于人脸的识别等。

3.1.4　身份认证技术分类

可以根据不同的分类标准对身份认证技术进行分类。

从是否使用硬件,身份认证技术分为软件认证和硬件认证。

- 软件认证是指在身份验证过程中不使用实体硬件,用户的身份验证信息依赖于各类软件。例如常用的动态密码保护程序,在身份认证过程中,会通过密保软件生成一个动态密码,服务器通过软件生成的动态密码对用户身份进行鉴别。
- 硬件认证是指用户的身份验证信息与硬件相关联,在身份验证过程中需要用到实体硬件。常用的认证硬件有磁卡、IC 卡、USB 令牌、其他硬件令牌等。

从认证需要验证的条件来看,身份认证技术分为单因子认证和双(多)因子认证。

- 单因子认证是指用户仅使用一个标识信息来验证自己的身份,静态密码就是一个典型的单因子认证方式。
- 双因子认证就是在单因子认证的基础上结合第二种认证因素的双重认证机制,从而进一步加强认证的安全性。目前使用最为广泛的双因子认证方法有动态口令牌+静态密码、USB Key+静态密码、二层静态密码等,其身份认证安全性远远高于单因子认证。

从认证信息来看,身份认证技术分为静态认证和动态认证。

- 静态认证是指在用户登录系统验证身份信息时,用户给服务器发送的身份认证信息是静态的、固定不变的。例如,采用静态口令认证机制,用户发送给服务器的认证信息是一串固定不变的静态密码。
- 动态认证是指用户登录系统验证身份过程中,发送给服务器的认证信息是动态变化的。典型的动态认证方式是如动态口令,这种认证机制会使用户密码随着时间或者使用次数而不断变化,而且每个密码只能使用一次。

从需要认证的对象可以分为单向认证、双向认证和第三方认证。

- 单向认证是指通信的双方只需要一方被另一方鉴别身份，例如常见的口令核对方式，当用户访问某台服务器时，单向认证只是由用户向服务器发送自己的身份信息，然后服务器对其进行比对检验，鉴别用户的身份真实性。
- 双向认证是指通信双方需要互相认证鉴别各自的身份。这主要应用在对安全性要求很高的系统中，例如网上银行系统，一方面银行网站要对用户身份进行认证，另一方面用户也需要鉴别银行网站的真实性。
- 第三方认证是指服务方和用户方的身份鉴别通过第三方来实现。每个用户都把自己的身份验证信息发送给可信第三方，由第三方负责认证过程。

3.2 常用网络身份认证技术

身份认证技术是在计算机网络中为完成确认操作者身份的过程而采用的技术方法。

3.2.1 口令认证

口令俗称密码，口令认证（也称为用户名（ID）＋密码（Password））广泛应用于计算机系统和日常生活中，是基于用户知道的信息进行认证的方法。每个用户的用户名和密码可以由用户自己设定，也可以由系统通过某些渠道（电子邮件、邮寄等）提供给用户，只有用户自己才知道，所以只要能够正确输入用户名和密码，系统就认为用户是合法的。

3.2.1.1 静态口令

常用的口令认证依机制是依靠静态口令（也称为可重用口令）来鉴别用户身份的合法性。系统为每一个合法用户建立一个用户名/口令对，当用户登录系统或使用某些功能时，提示用户输入自己的用户名/口令对（这些用户名/口令对在系统内是加密存储的），如果与某一项用户名/口令对匹配，则该用户的身份得到了认证。具体认证过程如图 3.1 所示。

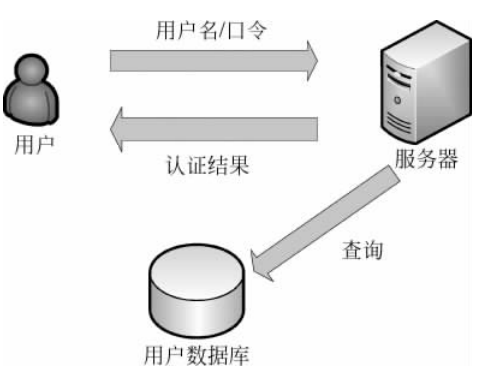

图 3.1 静态口令认证过程

静态口令认证的优点在于：基本上所有的计算机系统（如 UNIX、Windows、Linux 等）都支持对用户的口令认证，认证方式简单，易于实现。但这种认证方式的安全性较低，口令容易泄露。造成口令泄露的主要原因如下：

- 人为失误。比如无意中被他人看到，记录口令的载体丢失或被窃取等。
- 口令在用户端被截获。用户在访问系统的过程中以明文的方式输入口令，很容易被驻留在系统中的木马程序或网络监听设备截获。
- 口令在传输过程中被窃取。许多网络通信协议（如 FTP、HTTP、Telnet 等）都采用明文传输，这就意味着攻击者比较容易窃取传输过程中的认证信息，从而获得用户口令。
- 口令在系统端被截获。用户口令通过文件形式存储在系统端，这就使得攻击者可以利用系统漏洞截获用户口令。
- 字典穷举和猜测攻击。很多用户为了防止遗忘口令，通常采用一些有特定意义的字符串作为口令（如人名、电话号码、生日等），这些口令一般较短，攻击者就会将字符串的全集作为字典，对用户口令进行穷举攻击和猜测攻击。
- 伪造服务器攻击。由于许多系统只能进行单项认证，即系统能够认证用户，而用户无法对系统进行认证，这就使得攻击者可以伪造服务器来骗取用户的认证信息，进而获得用户的口令信息，这种攻击也称为网络钓鱼。
- 跨级别重复口令攻击。用户在访问多个不同安全级别的系统时，为了避免遗忘，经常采用相同的口令进行登录。低安全级别系统的口令比较容易被攻击者获得，从而对高安全级别系统进行攻击。
- 系统内部人员泄露。系统的内部人员能够通过合法途径获取用户的口令信息并非法使用。

为了提高口令认证的安全性，网络系统需要对口令信息进行安全加密存储和传输，限制账号登录次数，禁止共享账号和口令，设计或采用安全的口令认证协议，等等。

另外，用户要避免使用弱口令，具体要求如下：

- 口令的长度应至少为 8 个字符以上。
- 口令字符应由大小写英文字母、数字、特殊字符组合而成。
- 口令不能与账号名称相同。
- 不能用生日、电话号码和其他一些常用词等容易被猜测到的字符串作为口令。
- 所选口令不能包含在黑客攻击的字典库中。
- 避免使用系统默认口令。
- 经常更改口令，口令应有时效性。

3.2.1.2　动态口令

动态口令也称为一次性口令（One Time Password，OTP），这项技术是一种让用户密码按照时间或使用次数不断变化、每个密码只能使用一次的技术。用户进行认证时，除输入账号和静态口令之外，还必须输入动态密码。动态口令认证技术被认为是目前能够最有效地解决用户的身份认证安全性的方式之一，可以有效防范黑客木马盗窃用户账户口令、假网站等多种网络安全问题。

动态口令从生成方式上分为挑战/响应认证、时间同步认证、事件同步认证 3 种。

（1）挑战/响应（Challenge/Response）认证。

认证过程如下：

第一步，用户向系统发出认证请求。

第二步，系统产生一个随机数发送给用户，用户将这个随机数作为客户端验证算法的输入，此为挑战。

第三步，客户端将验证算法的输出（假设为 X）发送给系统，此为响应。

第四步，系统按照同样的算法计算出一个结果（假设为 Y），然后将用户发送来的 X 和 Y 进行比较，从而验证用户的身份。

由于验证算法只在客户端和系统端进行运算，不经过网络传输，提高了安全性。另外，针对用户的每一次认证请求，系统都会产生一个随机数给用户，所以每次的认证信息都不同，即使被外人截获也不会带来安全上的问题。

（2）时间同步（Time Synchronous）认证。

这种方式是以客户端和服务器端的同步时间作为认证的随机因素。客户端和服务器端都以用户登录时的时间作为验证算法的输入。系统将用户发送来的认证信息与本地验证算法运算的输出进行比较，从而完成认证。

这种方式对双方的时间同步要求较高，通常要求客户端时间与系统时间误差不超过60s，否则需要与服务器对时以保持同步。

（3）事件同步（Event Synchronous）认证。

这种方式以挑战/响应方式为基础，双方根据相同的前后相关的事件序列产生一系列的动态密码，然后进行比对验证。由于客户端可能会产生几组密码从而造成与系统的不同步，所以系统要能自动重新同步到目前使用的密码，一旦一个密码被使用过后，在密码序列中所有这个密码之前的密码都会失效。

事件同步认证的优点是认证卡容易使用；事件同步是唯一可以在批次运行环境下使用的技术，因为可以预先产生未来预计要使用的密码；由于使用者无法知道序列数字，所以这种认证方式安全性高，序列号码绝不会显示出来。

根据口令生成终端可以将动态口令分为手机令牌、短信密码、硬件令牌、智能卡等，其中手机令牌和硬件令牌统称为动态令牌。下面介绍主流的短信密码和动态令牌。

（1）短信密码。

短信密码属于手机动态口令的形式。身份认证系统以短信形式发送随机的6位或8位口令到用户的手机上，用户在登录或者交易认证时输入此动态口令，从而确保系统身份认证的安全性。短信密码由于其安全性、普及性、易收费、易维护等优点，被广泛应用于电子商务、银行金融、第三方支付等领域。

（2）手机令牌。

手机令牌也称手机口令牌，是用来生成动态口令的手机客户端软件。

手机作为动态口令生成的载体，在生成动态口令的过程中不会产生任何通信及费用，不会在通信信道中被截取，欠费和无信号对其不产生任何影响。由于其具有高安全性、零成本、无须携带、无物流等优势，相比硬件令牌其更符合互联网的精神。

手机令牌实质上是把动态密码技术用手机软件的方式实现。软件启动后，会运算产生一个不可猜测的动态密码，而且该软件可以运行在 Android、iOS、Symbian 等手机操作系统中。因此，手机令牌已经成为 3G/4G 时代动态密码身份认证令牌的主流形式。

（3）硬件令牌。

当前最主流的硬件令牌是基于时间同步的，动态口令是根据专门的密码生成算法每隔

60s 生成一个与时间相关的、不可预测的随机数字组合(通常为 6 位或 8 位),每个口令只能使用一次,每天可以产生 43 200 个密码。图 3.2 是硬件令牌的实物。

动态口令作为最安全的身份认证技术之一,目前已经被越来越多的行业所采用。其最大的优点在于,用户每次使用的口令都不相同,即使黑客截获了一次密码,也无法利用这个密码来仿冒合法用户的身份。但动态口令认证技术仍然存在用户操作烦琐(每次都要输入不同的口令密码),服务器端和客户端的时间要保持同步等问题。

图 3.2　硬件令牌

3.2.1.3　图形密码认证

传统的口令认证技术是依据用户提交的用户 ID 和相应的文本口令,这种字符式口令存在诸多缺点。图形密码使用图形作为认证媒介,通过用户对图形的单击、识别、重现或者用户与图形系统的互动进行认证。科学研究表明,人们对图形的记忆能力明显优于对文字的记忆能力,并且,随着图形数量的增多,图形密码的密钥空间要远大于文本密码,因此,其安全性也高于文本密码。

根据图形密码认证的实现方式不同,可以将图形密码分为两类:基于识别型和基于回忆型的图形密码。

基于识别型的图形密码身份认证要求用户记忆预先选定的一些特定图形,在认证阶段,系统会随机产生一组图形,让用户从中选出预先设定的图形,从而实现身份认证。

基于回忆型的图形密码身份认证则要求用户重复以前设定图形的过程。例如,在一种基于回忆型的图形密码身份认证方法中,在设定密码阶段,系统会要求用户在平面栅格上绘制出图形口令。在验证阶段,系统会显示同样的栅格,要求用户重复原来的设定过程,如果用户能够按照预定的规则绘制图形则通过验证。图 3.3 为目前智能手机、电子产品等使用较为广泛的一种基于回忆型的图形密码。

图 3.3　一种基于回忆型的图形密码

3.2.2　IC 卡认证

IC 卡(Integrated Circuit Card,集成电路卡)认证属于基于用户所拥有的实物进行鉴别的机制。IC 卡是一种内置集成电路的芯片,芯片中安全存储了与用户身份相关的信息。IC 卡由专门的厂商通过专门的设备生产,是不可复制的硬件。IC 卡认证技术广泛应用在现今社会的各个方面,例如第二代身份证、各地的市民卡、医疗卡、公交卡等。

IC 卡由合法用户随身携带,登录时必须通过专用的读卡器读取其中的信息,以验证用户的身份,只有持卡人才能被认证。

IC 卡认证通过 IC 卡硬件不可复制的特性来保证用户身份不会被仿冒。然而由于每次从 IC 卡中读取的数据是静态的,通过内存扫描或网络监听等技术还是很容易截取用户的身份验证信息,所以需要智能卡具备对信息加密的功能。另外,IC 卡认证还存在一个缺陷,就是系统只认卡不认人,而智能卡可能丢失,拾到或窃得智能卡的人将很容易假冒原持卡人的身份。

为了解决上述问题，可以综合前面提到的两类方法，实行双因子认证。即在进行认证时，既要求用户输入一个口令，又要求使用 IC 卡。这样，只要口令和卡不同时被其他人获取，用户就不会被冒充。

3.2.3 基于生物特征的认证

3.2.3.1 生物特征识别的概念

生物特征认别(Biometrics)就是指利用人的独一无二、可靠、稳定的生物特征来验证用户身份。生物特征是指可以测量或可自动识别和验证的唯一的生理特征或行为方式。生物特征分为身体特征和行为特征两类。常见的被用来进行身份验证的身体特征有指纹、视网膜、虹膜、掌型、脸型、人体气味、血管和 DNA 等，行为特征有语音、笔迹、击键特征、行走步态等。当前，对生物特征识别的研究方兴未艾，并且在许多场合(如机场、大型集会)的安保系统中已有应用，起到了重要的作用。从理论上说，生物特征认别是最可靠的身份认证方式，因为它直接使用人的物理特征来表示每一个人的数字身份，不同的人具有不同的生物特征，几乎不可能被仿冒。另外，基于生物特征的认证避免了其他认证方法中存在的遗忘、信息泄露、硬件丢失等现象。

能被用来作为身份识别的生物特征需要具备以下条件：

- 普遍性，即每个人都应该具有这一特征。
- 唯一性，即每个人在这一特征上有不同的表现。
- 稳定性，即这一特征不会随着年龄的增长和时间的改变而改变。
- 易采集性，即这一特征应该是容易测量的。
- 可接受性，即人们是否接受这种生物识别方式。

3.2.3.2 常见的生物特征识别技术

生物特征识别系统一般都包括对生物特征的采集、解码、比对和匹配过程。关键在于如何表示和采集这些生物特征，并将之存储于计算机中，以及如何利用有效、可靠的比对算法来完成用户身份的验证。

1. 指纹识别

指纹识别(fingerprint recognition)是目前应用最为广泛，比较成熟的生物识别技术。世界各地纷纷建立了指纹鉴定机构，成为司法刑侦中有效的身份鉴定手段。

指纹识别处理包括指纹图像采集、指纹图像处理特征提取、特征值的比对与匹配等过程。对指尖的纹线进行绘图，就能生成指纹。指纹扫描器能够读取指纹并将其转换成数字形式，这些数字副本可用来与存储在集中计算机系统中的经过授权的副本进行对比。图 3.4 为指纹识别过程。

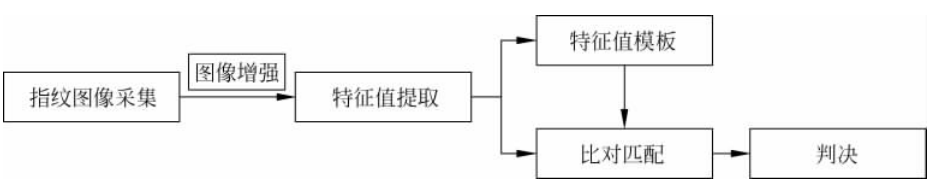

图 3.4 指纹识别过程

基于指纹的身份识别具有以下优点：

- 独特性。每个人的指纹具有唯一性。从几何特征到模式和纹线大小，每个指纹都有所不同。每个指纹一般有 70～150 个基本特征点，从概率学的角度，在两枚指纹中只要有 12～13 个特征点吻合，即可认定为同一指纹，按现有人口计算，起码要 120 年才能出现两个完全相同的指纹。
- 稳定性。一般人的指纹在出生后 9 个月得以成形并终身不变。
- 方便性。目前已有标准的指纹样本库，便于识别系统的软件开发；另外，识别系统中完成指纹采样功能的硬件部分（即指纹采集仪）也较易实现。
- 安全性。研究表明指纹识别对人体不构成侵犯。

但是，指纹识别技术也存在一些缺陷。例如，因为系统不能确定一个指纹是来自活体还是来自一个副本，可能受到欺骗。另外，受扫描装置或手指污渍的影响会降低指纹识别的可用性和方便性。

2. 掌纹识别

每个人的手的形状在人达到一定年龄之后就不再发生显著变化，而且都不同。掌纹识别就是利用手指和指关节的形状和长度等特征进行身份鉴定。

3. 视网膜识别

视网膜认证是根据人眼视网膜中的血管分布模式的不同来进行身份鉴别的。人眼球视网膜的中央动脉，在眼底至视神经乳头处分为上下两支，然后在视网膜颞侧上下及鼻侧上下再分为 4 支小动脉，各支小动脉再逐级分得更细、更小，以至在视网膜上形成四通八达的毛细血管网。

研究表明人眼视网膜中的血管分布具备唯一性特征，且在健康状况下非常稳定。但是，视网膜的采样较难，还没有标准的视网膜样本库供系统软件开发使用，这些问题导致视网膜识别系统在目前阶段难以开发，可行性较低。

4. 虹膜识别

人眼虹膜位于眼角膜之后，水晶体之前，其颜色因含色素的多少与分布不同而异。圆盘状的虹膜以中央的瞳孔为中心，其周围有辐射状的纹理和小凹。每个人虹膜的结构都不相同，并且这种独特的虹膜结构在人的一生几乎不发生变化。科学研究表明，世界上两个指纹相同的概率为 1/109，而两个虹膜图像相同的概率是 1/1011。因此，虹膜识别的错误率是各种生物特征识别中最低的。

虹膜识别技术也有很多地方有待完善：当前的虹膜识别系统只是用统计学原理进行小规模的实验，而没有进行现实世界的唯一性认证实验，而且虹膜图像获取设备相当昂贵。

5. 人脸识别

人脸识别（face recognition）是根据人脸各部分，如眼睛、鼻子、唇部、下颚等器官的相互位置，以及它们的形状和尺寸来区分人脸。图 3.5 是人脸识别的实例。

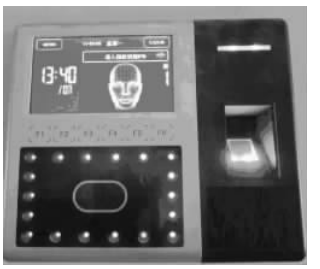

图 3.5　人脸识别

人脸识别系统主要包括 4 个组成部分：

- 人脸图像采集及检测。
- 人脸图像预处理。
- 人脸图像特征提取。
- 匹配与识别。

与基于指纹的人体生物识别技术相比，人脸识别是一种更直接、更方便、更友好、更容易被人们接受的识别方法。由于人的脸相会随年龄变化而变化，而且容易被伪装，所以人脸识别不是特别可靠。

6. 语音识别

语音识别(voice recognition)是基于人的声音特征(如频率)进行身份鉴定的。语音识别与指纹识别类似，每个人的语音特征具有唯一性。但是人的声音会随着年龄的增长或身体的健康因素而发生较大的变化。

7. 击键识别

击键识别(typing biometrics)属于人的行为特征识别，检查的是计算机用户的击键特征，包括速度、方式、力度、击键持续时间、击键间隙反应时间(前一次击键与后一次击键之间的延迟)等。一般来说，击键识别技术需要与其他认证方式相结合，比如在用户输入登录口令时发现有背离参考数据的情况出现，系统应该要求和允许用户通过其他认证技术实现认证。

8. 笔迹识别

笔迹(签名)识别也称为签名力学辨识(Dynamic Signature Verification，DSV)，它不是对签名图像本身的分析，而是通过对用户签名时的速度、加速度、笔压力及笔画长度等特征的分析来鉴别用户签名。

笔迹属于人的一种行为特征，笔迹的获取具有非侵犯性(或非接触性)，易被人接受。但是人的笔迹往往会有变化，身体状况和情绪变化也会影响到笔迹。此外，经过专门训练的人可以对笔迹进行模仿。这些都增大了笔迹识别的难度。

9. DNA 识别

DNA 是包含一个人所有遗传信息的片段，与生俱来，并终身保持不变。这种遗传信息蕴含在人的骨骼、毛发、血液、唾液等所有人体组织器官中。近年来，科学家们开发出多种 DNA 遗传标记用于个体识别。人的 DNA 图谱完全相同的概率仅为三千亿分之一。因此，通过 DNA 识别可以提供比较可靠的身份识别。

3.2.3.3 生物特征识别认证小结

随着社会对网络安全越来越重视，基于生物特征识别的身份认证技术也越来越受到重视，因为与传统身份认证技术相比，生物识别技术具有以下特点：

- 安全性更高。每个人拥有的生物特征各不相同，人的生物特征是个人身份的最好证明，满足更高的安全需求。
- 稳定性好。指纹、虹膜等人体生物特征不会随年龄等条件的变化而变化。
- 使用方便。每个人都具有自身独特的生物特征，用户不需要记忆密码和携带硬件(如 IC 卡)。

在设计或评价一个生物特征识别系统时，还要考虑以下几个方面：

- 易采集性。选择的生物特征易于测量，便于用户使用。
- 易接受性。选择的个人生物特征在采集时尽量减小对用户的侵犯，使用户更愿意接受。
- 可行性。包括对系统资源的要求、数据获取和分析的速度、识别的精确性和抗攻击能力。
- 性价比。针对实际的应用需求平衡软硬件和系统维护费用与性能。

本节所述的各种生物特征识别技术各有优劣，有各自的适用范围，有些技术还不够成熟，准确性和稳定性有待提高，还存在实施成本高的缺点。另外，生物特征识别是建立在假设从生物特征识别装置到认证系统的过程中是完全安全的基础上的。如果生物特征识别信息在网络传输过程中被获取，那么就面临身份假冒攻击的危险。但是，随着计算机性能的不断增强和模式识别、图像处理等技术的不断完善，将基于生物特征的身份识别技术融合在网络安全策略设计中将得到推广，从而大大增强网络的安全性。在对安全有严格要求的应用领域中，往往需要结合多种生物特征来实现更高精度、更可靠的身份识别系统。

3.3　网络身份认证协议

网络环境下的身份认证一般通过某种身份认证协议来实现。身份认证协议一般基于密码相关技术实现，定义了参与认证服务的各通信方在身份认证过程中需要交换的所有消息的格式、这些消息发生的次序以及消息的语义。

基于密码学原理的身份认证协议能够提供更多、更安全的服务。各种密码学技术都可以用来构造网络身份认证协议，按照所采用的密码技术的不同通常分为基于对称密码技术的认证和基于非对称密码技术的认证两种。

3.3.1　密码技术简介

随着计算机网络的发展，密码技术成为网络与信息安全的关键技术之一，是数字签名、数字证书和公共密钥基础设施(PKI)等安全措施的基础。

在密码技术中，将需要存储或者传输的原始数据称为明文(plaintext)，加密之后的数据称为密文(cipher)，密文是无序的数据，其内容无法理解。加密(encryption)是将明文经过编码使其转化为密文的过程，解密(decryption)是将密文还原为明文的过程。

加密和解密过程中使用的算法称为密码算法，是一个以加密/解密密钥(key)为参数的函数。密钥是二进制数的变量，用比特作为其长度单位，密钥越长越不容易被“破解”。

在现代密码学研究中，加密和解密算法一般都是公开的，任何人只要获知了密钥就能对密文进行解密，所以，密钥的设计与保护成为防范攻击的重点。根据所用密钥的不同，密码技术通常分为对称密码技术和非对称密码技术两种。

1. 对称密码

对称密码(symmetric key cryptography)也称作私钥密码，其加密和解密采用相同的密钥。发送者和接收者在进行安全的通信之前必须共享相同的密钥。

对称密码技术从加密模式上可分为两类：

- 流(stream)加密。对明文数据进行逐比特位加密得到密文。
- 块(block)加密。将明文分成固定长度的块(如 64 位一块)，用同一密钥和算法对每一块加密，输出固定长度的密文。

对称密码算法的处理速度通常要比非对称密码算法快。但是，对称密码算法的安全性取决于密钥的安全性，任何持有密钥的人都能够加密和解密消息。所以，对密钥的管理和传输的安全性要求较高。

对称密码中最常见的算法有 DES、IDEA、3DES、AES(Advanced Encryption Standard，高级加密标准)。后面要介绍的 Kerberos 身份认证系统就采用了 DES 算法。

2. 非对称密码技术

非对称密码(asymmetric key cryptography)中加密和解密采用一对不同的相关的密钥。每个通信方均需要有两个相关的密钥，通常将加密密钥公开，称为公钥(public key)，而解密密钥要求保密，称为私钥(private key)，所以也称为公共密钥密码(public key cryptography)技术。

由于非对称密码技术中不需要传输共享密钥，所以减少了密钥泄露的可能性。另外，由于每一对通信双方采用了不同的私钥，就算某个私钥泄露了，其他通信对的安全也不会受到影响。

非对称密码技术的复杂度要高于对称密码系统，速度为对称密码技术的 1/100～1/1000。所以，常用它来对少量关键数据进行加密，或者用于数字签名。例如，将非对称密码技术与对称密码技术相结合，即用非对称密码在通信双方之间传送对称密钥，用对称密码对实际传输的数据进行加密、解密。

应用最广泛的非对称密码算法是 RSA(由 Rivest、Shamir 和 Adleman 提出的并以他们的名字首字母命名)，典型的应用有安全套接字层(Secure Socket Layer，SSL)协议。其他还有 ElGamal、DSS 和 Diffie-Hellman 等算法，这些算法的复杂度和提供的功能各不相同。ElGamal 和 DSS 算法实现签名但是没有加密；Diffie-Hellman 算法用于建立共享密钥，没有签名也没有加密，一般与对称密码技术结合使用。

3.3.2 对称密码认证

3.3.2.1 概述

传统的基于用户名/口令的身份认证方式是对用户提交的用户名/口令进行验证，而用户名/口令在传输过程中可能会发生泄露。基于挑战/响应的技术可以实现既能够对用户所拥有的秘密信息(如口令)进行验证，又不会发生泄露。

但是，在网络环境中，一台计算机(如服务器)需要与很多用户进行身份认证，如果为每个用户都建立共享密钥，则增加了密钥创建、维护和更新的复杂性，同时降低了安全性。1978 年 Needham 和 Schroeder 提出了密钥分发中心(Key Distribution Center，KDC)的概念，KDC 与每个网络通信方都有一个共享密钥，并且被通信各方所信任。每对通信方之间的认证都借助于 KDC 这个可信第三方完成。KDC 负责为通信双方创建并分发共享密钥，通信双方获得共享密钥后再利用挑战/响应方式建立信任关系。

Kerberos 是由美国麻省理工学院(MIT)开发的一个认证协议,得到了广泛的使用,Kerberos 版本 5 已被 Internet 工程任务部(IEIF)正式接受为 RFC 1510,成为网络通信中身份认证的事实标准。Kerberos(或 Cerberus)原意是古希腊神话中的一种有 3 个头的凶猛的狗,是地狱的门卫。

Kerberos 的基本原理是:利用对称密码(DES 算法),通过可信的第三方(KDC)对网络上通信的实体进行相互身份认证,并在用户和服务器之间建立安全信道,能够阻止旁听和重放等攻击。其基本理念就是:如果通信双方都知道密钥,双方就可以通过确定对方知道密钥来相互确认身份。

一个 Kerberos 系统涉及以下一些基本实体和概念:

- 客户端(Client):用户用来访问服务器的设备。
- 目标服务器(Target Server):用户请求的应用服务器。
- 认证服务器(Authentication Server,AS):为用户分发票据授权票据(Ticket Granting Ticket,TGT)的服务器。用户使用 TGT 向票据授权服务器(Ticket Granting Server,TGS)证明自己的身份。
- TGS:为用户分发到目的应用服务器的票据(Ticket)的服务,用户使用这个票据向自己要求提供服务的服务器证明自己的身份。
- 密钥分配中心:通常将 AS 和 TGS 统称为 KDC。
- 领域(Realm):KDC 自治管理的计算机和用户等通信参与方的全体称为领域。领域是从管理角度提出的概念,与物理网络或者地理范围等无关。在实际使用中,为了方便,通常选择与 Internet 域名系统一致的名字来命名领域。不同领域中的用户之间也能进行身份认证。

此外,还有保证票据、密码等信息安全传输所需要的密钥。

3.3.2.2　Kerberos 的认证过程

当一个用户需要访问一个应用服务器时,它首先需要向目标服务器验证自己的身份,同时也要确认该服务器的身份,这就构成了双向身份认证。认证的步骤如图 3.6 所示。

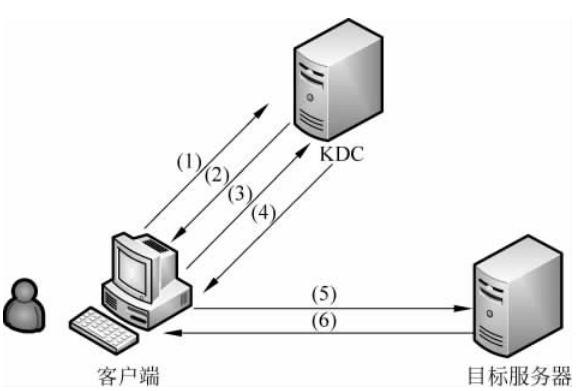

图 3.6　Kerberos 认证过程

(1) 客户端向 KDC 发送自己的身份信息(用户名/口令、IP 地址等),申请 TGT。

(2) KDC 根据收到的客户端发送来的信息进行认证,确认后从 AS 生成 TGT,并用事

先确定的客户端与KDC之间的共享密钥对TGT进行加密,然后回复给客户端。TGT包含客户端信息、时间戳、生存期等信息。此时只有真正的客户端才能利用它与KDC之间的共享密钥对TGT进行解密,从而获得TGT。共享密钥通常是用户口令经过哈希(Hash)生成的。

(3) 客户端再将获得的TGT和要请求的目标服务器等信息经过加密后发送给KDC,申请访问目标服务器所需的票据。

(4) KDC中的TGS生成一个会话密钥(Session Key),用于目标服务器对客户端的身份识别。然后KDC将这个会话密钥和用户名、用户地址(IP)、服务名、有效期、时间戳等一起封装成一个票据,并用它和目标服务器之间的密钥对这个票据进行加密。同时,用它和客户端之间的密钥对会话密钥进行加密。最后,将加密后的票据和会话密钥一并返回给客户端。

(5) 客户端将收到的票据转发至目标服务器。由于客户端不知道KDC与目标服务器之间的密钥,所以它无法篡改票据中的信息。同时,客户端对收到的会话密钥进行解密,然后将自己的用户名、用户地址(IP)打包成身份认证者(Authenticator)信息,用会话密钥对其进行加密,一并发送给目标服务器。身份认证者信息的作用是防止攻击者将来再次使用同样的凭据。

(6) 目标服务器利用它与KDC之间的密钥对收到的票据进行解密,从而得到会话密钥、用户名、用户地址(IP)、服务名和有效期等信息。然后再用会话密钥对身份认证者信息解密,获得用户名、用户地址(IP)等信息,并将其与之前从票据中解密出来的用户名、用户地址(IP)等信息进行比较以验证客户端的身份。最后将验证结果发送给客户端,响应用户的请求。

3.3.2.3 Kerberos的特点

Kerberos协议是专为开放网络设计的,充分考虑到了信息在网络传输过程中可能遇到的被截取、修改和插入等安全威胁,其安全性经过了长期的实践考验,具有以下特点:

- 客户端与KDC,KDC与目标服务器之间在协议工作前就需要有各自的共享密钥。
- Kerberos协议借助对称密码技术DES进行加密和认证,在每个客户端和目标服务器之间建立会话密钥(双方使用的临时加密密钥),保证了传递的消息具备机密性(Confidentiality)和完整性(Integrity),但是不具备抗否认性。
- Kerberos协议要求用户经过AS和TGS两重认证,减少了用户密钥中密文的暴露次数,以减少攻击者对有关用户密钥中密文的积累。
- Kerberos协议中的票据具有时效性,存放于用户的信用缓存中。凭据在有效期后自动失效,以后的通信必须从KDC获得新的票据进行认证。比如,当断开或退出网络时,票据即到期。系统管理员可以根据管理的需要改变票据的有效期长短,一般默认时间是一天。
- Kerberos运用票据的时间戳来检测对证书的重放和欺骗攻击。重放就是截获信息并把截获的信息进行修改,然后把修改后的信息重新发送给等待接收信息的实体。
- Kerberos协议认证具有单点登录(Single Sign-On,SSO)的优点,只需要用户输入一次身份验证信息,就可以利用获得的有效期内的TGT访问多个服务。
- 由于协议中的消息无法穿透防火墙,所以Kerberos协议往往用于一个组织的内部。

Kerberos也存在不足之处。例如,Kerberos协议在很多地方都涉及时间,如票据的有效期、时间戳等,如果各主机的时间偏差较大,则Kerberos认证系统将会失效。所以需要在

系统设计时考虑到时间的偏差，可以采取某些方法来解决各主机节点时间同步问题。如果某台主机的时间被更改，那么这台主机就无法使用 Kerberos 认证协议。一旦服务器的时间发生了错误，则整个 Kerberos 认证系统将会失效。另外，采用时间戳的方式防止重放攻击的代价也较高。

3.3.3　非对称密码认证

非对称密码算法中，私钥是保密的，外人无法获知，所以私钥往往就代表了某个通信参与方的身份。基于非对称密码的身份认证协议中，用户通过证明他知道某私钥来证明自己的身份，而且不需要将自己的私钥传输给服务方。

采用非对称密码方式进行身份认证时，需要事先知道对方的公钥，虽然可以采取某些方法来保证公钥传输的安全性，但是如果每个通信参与方都需要存储其他所有用户的公钥，既增加了负担又不便于更新维护，而且每个通信方自己产生的私钥和公钥的可信度也不一样，所以需要一个可信的第三方来参与公钥分发。在实际网络环境中，非对称密码认证系统采用证书(Certificate)的形式来管理和分发公钥。证书将一个实体和一个公钥捆绑，并且其他实体能对这种绑定进行验证。证书由证书权威机构(Certificate Authority，CA)签发。CA 是大家都信任的机构，充当可信的第三方角色。前文所述的 KDC 和 CA 都充当了分发密钥的角色，它们各有优缺点。

非对称密码身份认证方式的安全性更强，但是计算开销大。当前更多的安全系统利用非对称密码进行认证和建立对称的会话密钥，利用对称密码进行大数据量传输的加密，例如 SSL 协议、PGP 等。

非对称密码认证的一个显著优点是：只要服务器认为提供用户证书的 CA 是可信的，就认为用户是可信的，所以非常适合电子商务类的业务需求，例如信用卡支付。服务方可根据用户 CA 的发行机构的可靠性程度来对用户进行授权。

3.3.3.1　PKI

1. 数字证书

1) 什么是数字证书

数字证书(Digital Certificate)也称为数字标识(Digital ID)，是用来标识网络用户身份信息的一种特殊格式的数据编码，是用户或机构在网络环境中的身份证，用以确保网络传输信息的机密性、完整性以及通信双方身份的真实性、不可否认性。

数字证书采用公钥密码体制，每个用户用各自的私钥进行解密和签名，用公钥进行加密和验证签名。当发送一份保密文件时，发送方使用接收方的公钥对数据加密，而接收方则使用自己的私钥解密。因为用户私钥仅为他本人所有，所以就产生了别人无法生成的文件，也就形成了数字签名。采用数字签名，一是能够保证信息是由签名者自己发送的，签名者不能否认或难以否认；二是能够保证信息自签发后到收到为止未被修改，即签发的文件是真实文件。

用户如何获得密钥对？一般情况下，当用户申请数字证书时，激活安全设置会为用户产生密钥对。为了安全，密钥对应当在本地产生并且私人密钥不能在网上传输。一旦产生密钥对，就应在 CA 登记自己的公共密钥，随后 CA 将数字证书发送给用户，以证实用户的公

共密钥及其他一些信息。

用户如何发现别人的公共密钥？用户可以通过电子邮件或 CA 提供的目录服务等方式获取其他用户的公钥。一般的目录服务都具备抗攻击能力，用户可以确信其上所列的公共密钥都是可信的。为了保证 CA 的公共密钥的安全，必须使用很长的公共密钥（如 1024 位），有时还需经常更换密钥。

图 3.7 X.509 数字证书基本格式

2）数字证书的格式

目前广泛使用的数字证书标准是 X.509 v3，如图 3.7 所示。该国际标准规定了证书的格式，并且规定了建立证书发放系统的一些模式。

其内容主要包括：

- 版本号。描述该证书的版本，这可以影响证书中所指定的信息，迄今为止，已定义的版本有 3 个。例如使用的是 X.509 版本 3，则值为 2。
- 序列号。由证书颁发者（CA）给该证书分配的唯一标识符。
- 签名。用于说明该证书使用的数字签名算法，由对象标识符和相关参数组成。例如，SHA1（Secure Hash Algorithm，安全哈希算法）和 RSA 的对象标识符就用来说明该数字签名是利用 RSA 对 SHA1 杂凑加密。
- 颁发者。证书颁发者标识，必须是非空的。
- 有效期。表示证书有效的时间段，以起始日期和时间及终止日期和时间表示，必须要说明。所选有效期取决于许多因素，例如用于签写证书的私钥的使用频率及愿为证书支付的金钱等。
- 主体。证书拥有者标识，此字段必须是非空的，除非使用了其他的名字形式。
- 主体公钥信息。包括主体的公钥和该密钥所属公钥密码系统的算法标识符及所有相关的密钥参数。
- 颁发者唯一标识符。属于可选项。
- 主体唯一标识符。证书拥有者的唯一标识符，属于可选项。
- 扩展。可选的标准和专用扩展。

3）数字证书的种类

根据使用者的不同，数字证书可以分为用户证书、系统证书、软件证书 3 种。用户证书为个人、机器或机构提供身份凭证；系统证书是指 CA 系统自身的身份凭证；软件证书通常为可以从网络上下载的软件提供凭证，以便下载用户获取相关信息。

4）数字证书的存储

数字证书的存储介质主要有硬盘、IC 卡及 USB Key 等形式。使用硬盘存储方式适用于不常更换计算机的个人用户，但这种方式存在一个安全隐患，因为在使用证书时必须将证书和私钥导入浏览器（如 IE）中，所以其他人可以通过使用用户的计算机以非法使用该用户的数字证书。使用 IC 卡和 USB Key 就可避免发生上述安全问题，因为用户私钥是在 IC 卡和 USB Key 中产生的，且私钥不可导出。在 IE 中使用导入的证书时，如果没有 IC 卡或 USB Key 也是无法使用的。由于 IC 卡必须要有专用的读写器，使用不太方便，因此小巧美

观、安全方便的 USB Key 逐渐成为数字证书存储的首选设备。

当前许多场合使用的是浏览器数字证书。浏览器证书存储于 IE 浏览器中，可任意备份证书和私钥。客户端不需要安装驱动程序（根据情况可能需要下载安装最新的签名控件），且无需证书成本。IE 浏览器证书比较适合有固定上网地点的客户，可以通过 IE 浏览器进行查看。

选择 IE 浏览器的“工具”菜单“Internet 选项”命令，在对话框中选择“内容”选项卡，如图 3.8 所示。

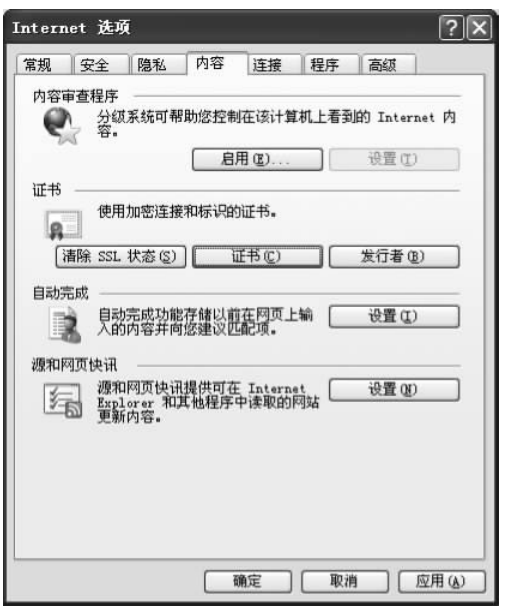

图 3.8　IE 的数字证书选项窗口

单击“证书”按钮，系统将弹出证书管理器窗口，如图 3.9 所示。

图 3.9　IE 证书管理器窗口

选择需要查看的证书，然后单击“查看”按钮，系统弹出证书查看窗口，如图 3.10 所示。窗口中显示了该证书的相关信息。

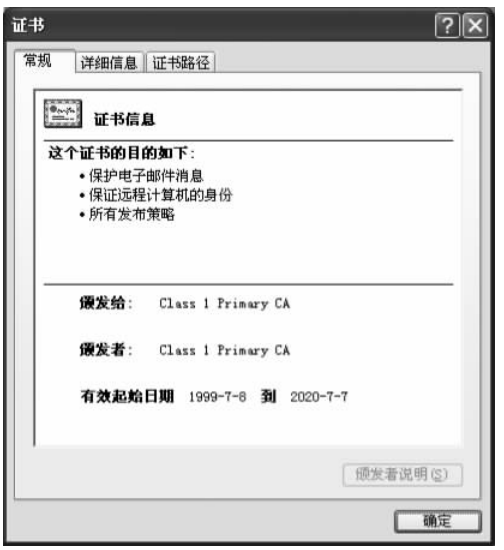

图 3.10 IE 证书查看窗口

单击图 3.9 中的“高级”按钮，可以查看证书的使用目的，如图 3.11 所示。

图 3.11 IE 证书目的

根据用途的不同，数字证书可以分为签名证书和加密证书两种。签名证书用于对用户传输的信息进行签名，数据接收方可以根据数字证书来确认发送方的身份。由于发送方的数字证书只有发送方具有，所以具备不可否认性特征。加密证书用于对用户传输的信息进行加密，只有正确的数据接收方才能对加密信息进行解密，而且可以判断传输的信息是否在传输过程中被篡改过，所以具备保密性和完整性特征。对于加密证书，CA 需要备份用户的私钥。

2. PKI 的定义

PKI(Public Key Infrastructure，公钥基础设施)是采用非对称密码(公钥密码)的原理

和技术建立的具有通用性的提供安全服务的安全基础设施，包括创建、管理、存储、分发和撤销公钥证书所需要的相关硬件、软件和策略。

PKI 采用证书管理密钥，通过可信 CA 将用户的身份信息与其公钥相捆绑，提供身份认证服务。PKI 提供了一种系统化的、可扩展的、统一的、容易控制的公钥管理和证书签发体系，通过各组件和策略组合为网络通信的机密性、完整性、真实性和不可否认性提供保障。

基于 PKI 的认证服务通过数字签名和密码技术来确认身份。假如实体 A 需要验证实体 B 的身份，那么首先 A 要获取 B 的证书，并用双方共同信任的 CA 的公钥验证 B 的证书上 CA 的数字签名，如果签名通过，则说明 B 的证书是可信的。然后，A 向 B 发出随机字符串信息，B 接收到信息后，用 B 的私钥进行签名处理后再发回 A。如果 A 能够利用 B 的证书解密 B 签名的信息，则 A 就确认了 B 的身份。这是因为只有 B 的公钥才能解开其签名的信息。

PKI 是当前互联网通信安全的重要技术和基础，为电子商务、电子政务等互联网应用提供安全保障。PKI 技术遵循相关的国际标准和 RFC 文档（如 PKCS、SSL、X.509、LDAP 等），提供了比较成熟、完善的网络系统安全解决方案。随着新的技术不断出现，CA 间的信任模型、使用的密码算法和密钥管理方案等越来越完善。

3. PKI 的组成

一个 PKI 系统需要多个组件实体之间的联合操作，主要包括认证中心（CA）、注册中心（Registration Authority，RA）、LDAP（LightWeight Directory Access Protocol，轻量目录访问协议）目录服务器、应用接口等，如图 3.12 所示。

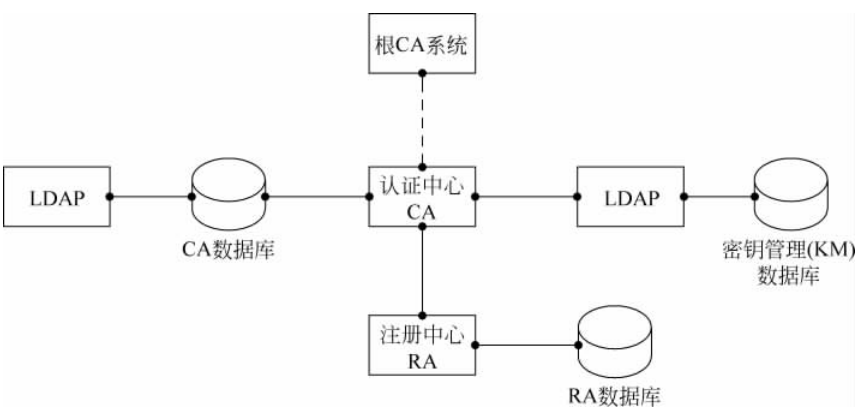

图 3.12　PKI 的组成结构

1）认证中心（CA）

CA 是整个 PKI 系统中的可信第三方，它保证了公钥证书的合法性，是整个 PKI 系统的核心，负责对用户证书的签发、作废、更新和管理。由于 CA 得到各方的信任，所以拥有它签发的数字证书的通信对方的身份也就可以信任。

2）注册中心（RA）

RA 负责对证书申请用户进行审查，对通过审核的用户进行注册，并协助 CA 对证书进行签发和管理。一些小规模的 PKI 系统中没有设立独立的 RA，其职能由 CA 担负，但这样会增加整个系统的安全风险。

3) LDAP 目录服务器

LDAP 目录服务器用于存取证书和证书作废表(Certificate Revocation List,CRL)信息。目录系统是 PKI 的重要基础,LDAP 协议是访问证书库和 CRL 的主要方式,是访问 PKI 目录服务的标准协议要求。用户可通过 LDAP 目录服务进行证书和公钥的查找和获取,通过查询 CRL 以验证用户的证书状态。

4) 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。

4. PKI 的功能

一个完整、有效的 PKI 系统功能主要包括注册管理、证书签发、证书作废、证书管理、证书校验、密钥管理等功能。

1) 注册管理

注册是即将成为证书主体的终端实体使 CA 认识自己的过程。终端实体可以通过 RA 注册,如果由 CA 实现 RA 的功能,终端用户也可以直接向 CA 注册。RA 主要负责对用户的身份信息进行收集和资格审查,主要包括以下几个功能:

- 获取用户身份信息。用户将个人身份信息(密码、E-mail 等)提交给 RA,RA 完成用户注册信息的填写。
- 审核用户信息。对用户的注册信息进行审核,审核通过后,产生用户的 PIN。PIN 是 RA 赋予用户的标识,所以要求 PIN 具有唯一性,另外 PIN 还应具备随机性特征和足够的长度以应对猜测攻击和穷举攻击。
- 注册。以用户的 E-mail 的哈希值作为密钥对 PIN 进行加密。保存用户的 E-mail、密码和加密后的 PIN,作为以后对用户身份进行验证的凭据。将加密后的 PIN 以安全的方式发送给用户。
- 提交证书生成申请。RA 向 CA 提交证书生成申请。

2) 证书签发

证书签发是 CA 乃至整个 PKI 的核心功能,主要包括以下步骤:

(1) 用户提交证书申请。如果用户申请的是加密证书,申请信息只有用户信息。如果申请的是签名证书,则申请信息中还要包含用户的公钥。

(2) RA 对申请进行审核。有的 PKI 系统需要 CA 进一步做审核。如果审核通过,RA 将向 CA 提交证书生成请求。

(3) CA 生成证书。如果生成的是加密证书,CA 需要产生一对公私钥,公钥用于备份用户的私钥,私钥用于恢复用户的私钥。如果生成的是签名证书,需要对用户数字签名进行验证。

(4) 证书发布。CA 在签发一份证书后,需要在系统内公布用户的证书,以便其他用户能获取。最常用的发布形式是将用户的证书存储到 LDAP 目录服务器上。也可以发布到 Web 服务器(返回给用户一个 URL,供用户下载)、FTP 服务器或其他目录访问服务器(比如 X.509)上。

(5) 用户下载和安装证书。用户下载个人证书,并安装到浏览器。在安装加密证书时

需要输入证书安装密码。如图 3.13 所示，中国银行的网银用户登录窗口右侧就提供了“CA 证书下载”服务。

图 3.13　中国银行网银登录窗口

3）证书撤销

在数字证书的有效期内，如果由于某些原因需要提前停止使用，证书就需要被撤销。例如，证书的一些信息（如用户名、单位等）发生了改变、私钥被泄露等。CA 在收到证书撤销申请后执行证书撤销，并通知用户。被 CA 作废的证书将不再可信，所以用户在使用证书时，系统需要检查证书是否已被撤销。

证书撤销的实现方法有两种：

- 利用周期性发布机制，典型的是证书撤销列表（CRL）。
- 在线证书状态协议（Online Certificate Status Protocol，OCSP）。

CRL 数据结构的内容包括版本号、签名算法标识符、发现者名称、本次发布时间、下次更新时间、撤销的证书（证书序列号、撤销时间）等。

CA 在撤销一个证书后就对 CRL 进行更新，增加被撤销证书的信息。CRL 的大小随着被撤销的证书增多而不断变大。对此有两种解决办法：一是采用分段式 CRL，将一个 CA 的撤销信息存放在多个 CRL 中，这些 CRL 可以分布地存放在多个服务器上；二是采用增量 CRL（delta-CRL）方式，基本思想是每撤销一个证书只产生新增加的证书撤销信息，用户通过获取增量 CRL 来更新本地的 CRL。

OCSP 为用户提供实时在线证书状态查询，这样可以避免由于 CRL 太大而造成的传输困难、处理效率低下的问题，也避免了 CA 中的 CRL 和用户的 CRL 不一致的现象，增强了

安全性。

4）证书管理

除了前面介绍的证书的发布和撤销外，证书管理包括的功能还有证书验证、证书更新、证书归档等。

（1）证书验证。

用户在对证书进行验证时需要完成以下任务：验证证书的签名，确定证书的合法性；检查证书的有效期；核实证书的用途是否符合要求；确认该证书没有被撤销。在一个复杂而庞大的 PKI 系统中，CA 具有层次结构或是分布式的，用户在对证书验证时需要进行证书链校验或交叉认证，具体内容在后面的信任模型部分详细说明。

（2）证书更新。

在证书已到有效期或者证书的一些属性已经改变且需要重新证明时需要进行证书更新。证书更新包括用户证书更新和 CA 证书更新两种。

用户证书的更新方式有两种：

- 人工更新，RA 根据用户的更新申请信息对用户证书进行更新。
- 自动更新，CA 对快要到期的用户证书自动进行更新。

由于 CA 证书的特殊性，需要采取一些步骤使得向新证书的转换更加平滑。CA 证书更新时要用它的新私钥为旧公钥签名，用旧私钥为新公钥签名，最后再用新私钥为新公钥签名，这时自签名的 CA 证书代表新的可信第三方。

（3）证书归档。

证书失效、撤销或者更新后需要存储旧的证书，也就是证书归档，以满足用户对历史信息的查阅和验证要求。因为用旧证书签名或加密的信息无法用新证书进行认证或解密，PKI 通过证书归档以保证安全服务的持续性。

5）密钥管理

在 PKI 系统中，密钥管理主要包括密钥生成、密钥备份和恢复、密钥更新、密钥销毁和归档处理等。

PKI 技术要求每个用户拥有两对公私密钥。其中一对用于数据加密和解密，另一对用于数字签名和校验签名，以支持数字签名的不可否认性。这两对密钥在管理上的要求并不一样。

（1）密钥生成。

用于加密/解密的密钥对可以在客户端生成，也可以在一个可信的第三方机构生成。如果在异地生成该密钥对，必须能够保证将其安全地传输到客户端供客户使用。

用于签名/校验的密钥对一般要求在客户端生成，特殊情况下（例如客户端没有能力生成密钥对）可以在一个可信的第三方生成。但是，该密钥对中用于签名的私钥只能由用户自身唯一拥有，严禁在网络中传输，或存放于网络中的其他地方。如果该密钥对是由第三方生成的，则在用户获得该密钥对后，第三方必须销毁其中的私钥。但用于校验签名的公钥可以在网络中传输，还可以随处发布。

（2）密钥备份和恢复。

PKI 要求应用系统提供密钥备份与恢复功能。当用户忘记密钥访问口令或存储用户密钥的设备损坏时，可以利用此功能恢复原来的密钥对，从而使原来加密的信息可以正确

解密。

并不是用户的所有密钥都需要备份，也并不是任何机构都可以备份密钥。可以备份的密钥仅限于用于加密/解密的密钥对，而用于签名/校验的密钥对则不可备份，否则将无法保证用户签名信息的不可否认性。用于签名/校验的密钥对在损坏或泄露后必须重新产生。可以备份密钥的应该是可信的第三方机构，如 CA、专用的备份服务器等。

(3) 密钥更新。

密钥的使用是存在有效期的。当密钥到期时，PKI 应用系统应该可以自动为用户进行密钥更新。也可以由用户主动向 RA 申请更新，同时进行证书更新。

(4) 密钥归档。

当用于加密/解密的密钥对成功更新后，原来使用的密钥对必须进行归档处理，以保证原来的加密信息可以正确地解密。但用于签名/校验的密钥对成功更新后，原来密钥对中用于签名的私钥必须安全地销毁；而原来密钥对中用于校验签名的公钥要归档管理，以便将来对旧的签名信息进行校验。

PKI 系统的密钥管理总体来说应该是自动的，并且是对用户透明的。有的 PKI 系统还要求能为一个用户管理多对密钥和证书，能够提供对密钥周期和用途等进行设置的安全策略编辑和管理工具。好的密钥管理能提高 PKI 系统的扩展性和降低运行成本。

5. 信任模型

通常一个 CA 为一个有限的用户团体提供服务，这样的用户团体通常被称为安全领域(Security Domain)。大型网络系统中往往存在多个 CA，所以 PKI 需要建立不同安全领域之间的相互信任关系。信任模型是 PKI 中建立信任关系和验证证书时寻找和遍历信任路径的模型。

1) 单 CA 信任模型

单 CA 信任模型是最基本的信任模型，即整个 PKI 系统中只有一个 CA。该 CA 为系统中所有用户提供安全服务，被所有用户所信任，如图 3.14 所示。

单 CA 信任模型容易实现，易于管理，只需要建立一个 CA，所有用户之间都能相互认证。但是，该模型对于拥有大量用户或不同的用户群体的系统支持困难。

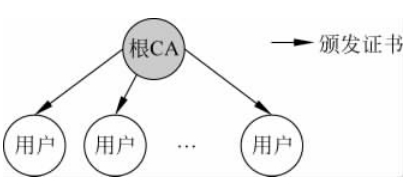

图 3.14　单 CA 信任模型

2) 严格层次信任模型

在严格层次信任模型中，通过 CA 间的主从关系建立信任模型。可以用一棵倒置的树对其进行描述，如图 3.15 所示。

这种模型中有一个特殊的 CA 称为根 CA，每个用户都知道根 CA 的公钥，所有用户都信任根 CA，根 CA 的证书由自己签发。根 CA 下可以有零层或多层子 CA，上层 CA 为下层 CA 签发证书，倒数第二层的子 CA 为以它为根的用户群体签发证书，通常其他层的 CA 不直接为用户签发证书。该模型中的信任关系是单向的，各级 CA 组成了一个信任链。两个用户进行相互认证时，双方都提供自己的证书和签名，通过根 CA 来对证书进行有效性和真实性的认证。

严格层次信任模型具有扩展性好的优点，比较容易增加新的信任域，而且证书路径长度一般不会很长。但是单个 CA 的失败会影响整个 PKI 体系，影响的大小与其离根 CA 的距离相关，根 CA 的失效将导致整个 PKI 系统的失效。

3）网状信任模型

网状信任模型又称为分布式信任模型，与严格层次信任模型相反，网状信任模型将信任分散到两个或多个 CA 上，如图 3.16 所示。

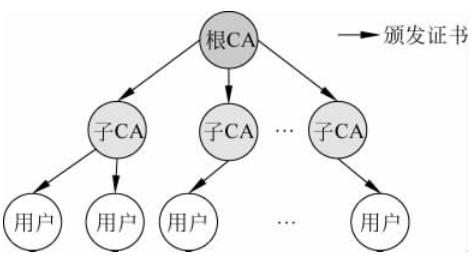

图 3.15 严格层次信任模型

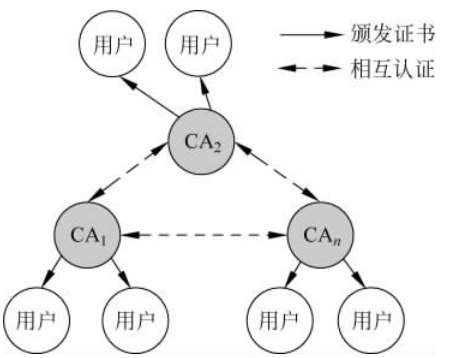

图 3.16 网状信任模型

如果任意两个 CA 间都存在相互认证，则这种模型成为严格网状信任模型。有的复杂系统中会结合网状模型与层次模型，建立混合型信任模型。

网状模型具有更好的灵活性，单个 CA 的安全性对整个 PKI 系统的影响有限。增加新的认证域也方便，只要新的 CA 与网中其他至少一个 CA 建立信任关系即可。但是，网状模型也存在认证路径发现难和实现复杂的缺点。

4）桥 CA 信任模型

桥模型被设计用来克服层次模型和网状模型的缺点和链接不同的 PKI 体系。桥 CA 通过分别与多个信任域的 CA 进行交叉认证的方式，建立不同信任域的 CA 之间的信任路径，从而实现不同信任域实体之间的互连、互通、互操作，允许用户保持原有的信任 CA，如图 3.17 所示。桥 CA 不同于树状结构和网状结构中的 CA，它不直接向用户签发证书，它也不像根 CA 那样是可信实体。如同网络中使用的集线器，任何结构类型的 PKI 都可以通过桥 CA 连接在一起，实现彼此间的信任。

桥 CA 的实用性很强，代表了现实世界中证书机构的相互关系，证书路径较易发现，路径较短。但桥 CA 存在证书路径的有效发现和确认困难、证书复杂、证书和证书状态信息获取困难、大型 PKI 目录的互操作性不方便的缺点。

5）以用户为中心的信任模型

在以用户为中心的信任模型中，每个用户自己决定信任哪些证书。用户自己就是自己的根 CA，没有可信的第三方作为 CA，如图 3.18 所示。

这种模型中用户的可控性很强。例如用户 A 收到一个标明是 B 的证书，但是发现该证书是由他不认识的 C 签名的，但是 C 的证书是由用户认识且信任的 D 签名的，于是就存在一个从 D 到 C 到 B 的密钥链。这时用户可以自我决定是否信任 B 的证书。

这种模型对用户自身的决策能力要求较高，所以一般适用于技术水平较高和利害关系

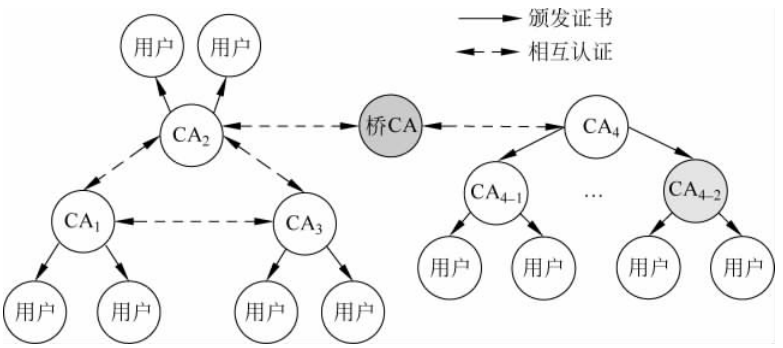

图 3.17　桥 CA 信任模型

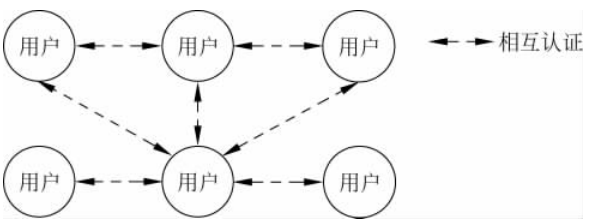

图 3.18　以用户为中心的信任模型

高度一致的群体中，不适用于金融或政府环境，因为这些环境通常是需要对用户的信任行为实行某种控制的。

6）Web 信任模型

Web 信任模型建立在浏览器的基础之上，浏览器中内置了多个根 CA，各个根 CA 间是相互平行的，浏览器用户信任这些根 CA，如图 3.19 所示。由于这些根 CA 是由浏览器厂商内置的，厂商隐含认证了这些根 CA，所以浏览器厂商是实际上的根 CA。

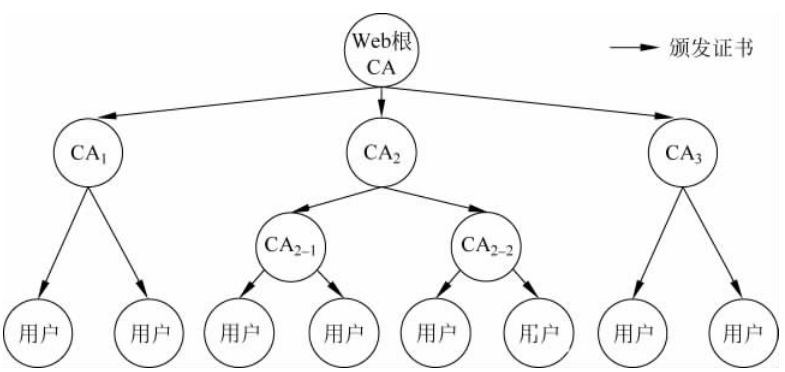

图 3.19　Web 信任模型

Web 信任模型操作性强，使用方便，对用户的要求较低。但是，存在安全性较差和根 CA 与用户的信任关系模糊的缺点。嵌入的多个根 CA 只要有一个失效，安全性也将被破坏，而且没有实用的机制来发现和撤销失效的根 CA。另外，用户很难知道某个浏览器嵌入了哪些根 CA，也无法知道这些根 CA 的依托方是谁。

6. PKI 相关的国际标准

与 PKI 相关的国际标准可以分为两类：一类用来定义 PKI，另一类依赖于 PKI。

1) 定义 PKI 的标准

在 PKI 系统中，用户的注册流程、数字证书的格式、CRL 的格式、证书的申请格式以及数字签名格式等都有相关的国际标准进行了严格的定义。

- X.509。由国际电信联盟 ITU 制定，用来对 PKI 中的数字证书进行规范化定义。
- PKCS。由美国 RSA 数据安全公司及其合作伙伴制定的一组公钥密码学标准，内容包括证书申请、证书更新、CRL 发布、数字签名、扩展证书以及数字信封的格式等方面的一系列标准。
- PKIX。由 IETF 组织中的 PKI 工作小组制定，主要定义了 PKI 系统中的用户、CA、RA 和证书存取库等的模型。

2) 依赖于 PKI 的标准

当前有很多依赖于 PKI 的安全标准，如安全的套接层协议(SSL)、传输层安全协议(TLS)、安全的多用途互联网邮件扩展协议(S/MIME)、IP 安全协议(IP Sec)等。

- S/MIME。是一个用于发送安全报文的 IETF 标准。它采用了 PKI 数字签名技术并支持消息和附件的加密，无须收发双方共享相同密钥，S/MIME 采用 PKI 技术标准来实现，并适当地扩展了 PKI 的功能。目前该标准包括密码报文语法、报文规范、证书处理以及证书申请语法等方面的内容。
- SSL/TIS。是互联网中访问 Web 服务器最重要的安全协议，也可以应用于基于客户/服务器模型的应用系统，SSL/TLS 都利用 PKI 的数字证书来认证客户和服务器的身份。
- IPSec。是 IETF 制定的 IP 层加密协议，采用了 PKI 中进行加密和认证过程的密钥管理的功能。IPSec 主要用于开发新一代的 VPN。

3.3.3.2 RADIUS 协议

1. AAA 简介

AAA 是 Authentication(认证)、Authorization(授权)和 Accounting(计费)的简称。这里的认证就是本章所指的对用户的身份进行验证，判断其是否为合法用户。授权是指当用户身份被确认合法后，赋予该用户能够使用的业务和拥有的权限，例如分配一个 IP 地址。计费是指网络系统收集、记录用户对网络资源的使用情况以便向用户收取费用和进行审计。AAA 是网络运营的基础，既保证了合法用户的权益，又有效地保证了网络系统的运行安全。

RADIUS(Remote Authentication Dial-In User Service，远程认证拨号用户服务)是使用广泛的用户接入管理协议。最初，Livingston 公司提出 RADIUS 协议的目的是简化认证流程，便于进行大量用户的接入验证。后来，经过不断扩充和完善，其应用范围扩展到无线验证和 VPN 验证等领域，提供成熟的 AAA 管理。

2. RADIUS 的工作过程

RADIUS 是基于 UDP 的应用层协议，认证使用 1812 端口，计费使用 1813 接口。

RADIUS 采用客户/服务器模式，其中客户端是指网络接入服务器(Network Access

Server，NAS)或 RADIUS 客户端软件，服务器端是指 RADIUS 服务器。

- 客户端的功能是把用户身份信息(用户名、密码)传输给 RADIUS 服务器，并处理返回的响应。
- RADIUS 服务器的功能是接收客户端发来的用户接入请求，对用户身份进行验证，以提示用户认证通过与否，是否需要 Challenge 身份认证，并返回给客户端为其提供服务所需的配置信息。

RADIUS 服务器采用数据库的形式中集中存放用户的相关安全信息，避免安全信息凌乱散布带来的不安全性，同时更可靠且易于管理。实施计费时，客户端将用户的上网时长、进出字节数、进出包数等原始数据送到 RADIUS 服务器上，以供 RADIUS 服务器计费时使用。

一个 RADIUS 服务器可以充当其他 RADIUS 服务器或其他模式的认证服务器的代理，以支持漫游功能。所谓漫游功能，就是代理的一个具体实现，可以让用户通过本来和其无关的 RADIUS 服务器进行认证。

RADIUS 认证授权工作的主要步骤如图 3.20 所示。

图 3.20　RADIUS 认证授权过程

(1) 用户首先启动与客户端的连接(例如采用 VPN 拨号、Telnet 等)，输入用户名和密码。

(2) 客户端采用非对称加密算法 MD5(Message Digest Algorithm 5，消息摘要算法第 5 版)对密码进行加密，再将用户名、密码、客户端 ID 和用户访问端口的 ID 等相关信息封装成 RADIUS“接入请求(Access Request)”数据包并发送给 RADIUS 服务器。

(3) RADIUS 服务器对用户进行认证，必要时可以提出一个 Challenge，收集用户的附加信息以进一步对用户进行认证。

(4) 如果用户通过认证，RADIUS 服务器向客户端发送“允许接入(Access Accept)”数据包。如果用户信息没有通过认证(用户名或口令不正确)，则向客户端发送“拒绝接入(Access Reject)”数据包，或者是发送“重新输入口令(Change Password)”数据包要求用户重新输入口令。

(5) 若客户端收到的是允许接入包，则向 RADIUS 服务器提出计费请求(Account Require)，RADIUS 服务器进行响应(Account Accept)，对用户的计费开始。同时，授予用户相应的权限以允许用户进行自己的相关操作。如果客户端收到的是拒绝接入包，则是拒绝用户的接入请求。

3. RADIUS 数据包格式

RADIUS 数据包格式如图 3.21 所示。

1) Code

Code 字段长度为 1B，用于区分 RADIUS 数据包的类型。常用的 Code 值(十进制)和

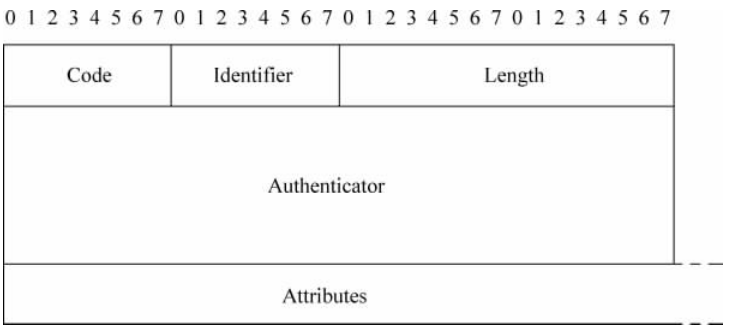

图 3.21 RADIUS 数据包格式

对应的数据包类型如下：

Code＝1，接入请求(Access-Request)。

Code＝2，接入允许(Access-Accept)。

Code＝3，接入拒绝(Access-Reject)。

Code＝4，计费请求(Accounting-Request)。

Code＝5，计费响应(Accounting-Response)。

Code＝11，接入询问(Access-Challenge)。

Code＝12，服务器状态(Status-Server(experimental))。

Code＝13，客户端状态(Status-Client(experimental))。

Code＝255，预留(Reserved)。

2) Identifier

Identifier 字段长度为 1B，用于请求和应答包的匹配，一般是短期内不重复的数值。RADIUS 服务器能检测出具有相同的客户源 IP 地址、源 UDP 端口及标识符的重复请求。

3) Length

Length 字段长度为 2B，用于表示 RADIUS 数据包(包括 Code、Identifier、Length、Authenticator、Attributes)的总长度，最小为 20B，最大为 4096B。数据包超出长度域所指示的部分将被看作是填充字节而被忽略(不接收)，如果数据包大小比长度域所指示的小，则必须丢弃该分组。

4) Authenticator

Authenticator 字段长度为 16B，用于验证 RADIUS 服务器的应答和对用户口令的加密。通过 RADIUS 服务器与客户端的共享密钥以及请求认证码(Request Authenticator)和应答认证码(Response Authenticator)共同支持发、收数据包的完整性和认证。

(1) 请求认证码。

在接入请求数据包中，请求认证码是一个 16B 的随机二进制数。在密钥的整个生存周期中，这个值应该是唯一且不可预测的，因为具有相同密钥的重复请求值会使黑客有机会使用已截取的响应回复用户。因为同一密钥可以用在不同地理区域中的服务器的验证中，所以请求认证码应该具有临时的全球唯一性。

在请求接入和请求计费数据包中的请求认证码的生成方式是有区别的。对于请求接入包，请求认证码是 16 个 8B 的随机数。对于计费请求包，认证码是一串由 Code、Identifier、

Length、16 个为 0 的 8B、请求属性和共享密钥所构成的字节流经过 MD5 加密算法计算出的散列值，即

Request_Auth = MD5(Code＋Identifier＋Length＋16 个为 0 的 8B＋Attributes＋Shared Secret)

(2) 响应认证码。

响应认证码是允许接入、拒绝接入、接入询问和计费响应数据包中的认证码值，它是一串由编码域、标识符、长度、来自接入请求数据包的请求认证码和执行共享机密的响应属性构成的字节流上计算出的单向 MD5 散列，即

Response_Auth = MD5(Code＋ Identifier ＋Length＋Request_Auth＋Attributes＋Shared Secret)

5) Attributes

Attributes 字段长度可变，由包含的属性的类型和长度决定。每个 RADIUS 数据包可以有 0 个或多个属性，RADIUS 协议通过不同的属性定义各种操作。不同的属性包含不同的信息，每个属性由 3 部分组成：类型(Type)、长度(Length)、属性值(Value)。用户可以根据实际需要在不中断已存在协议执行的前提下自行定义新的属性。有关属性的详细内容可以参看 RFC 文档。

4. RADIUS 认证的安全措施

1) 用户口令加密

采用 MD5 加密算法对客户端和 RADIUS 服务器之间传输的用户口令进行加密，防止口令泄露。

2) 认证机制

客户端和 RADIUS 服务器之间利用共享密钥技术和认证码方式进行认证，保证数据传输的完整性、机密性，同时防止网络上的其他主机冒充客户端或 RADIUS 服务器。具体实现过程如下：

(1) 客户端生成包括请求认证码的接入请求数据包，发送给 RADIUS 服务器。

(2) RADIUS 服务器收到客户端的接入请求后，根据用户名在数据库中查找匹配项。如果找到，则采用与客户端一致的方法也产生一个认证码。

(3) 如果两个认证码一致，则发送允许接入数据包给客户端。否则，发送拒绝接入数据包。

(4) RADIUS 服务器构造包含响应认证码的响应数据包，发送给客户端。

(5) 客户端收到认证响应数据包后，根据正在等待响应的那个请求的请求认证码和响应包的内容也产生一个响应认证码，将这个响应认证码与 RADIUS 服务器发送来的认证码相比较。若相等，则认证通过，建立连接，否则认证失败。

3) 用户与客户端之间的认证

RADIUS 协议可以支持多种用户与客户端之间的认证方式，例如 PAP(Password Authentication Protocol，密码认证协议)、CHAP(Challenge Handshake Authentication Protocol，挑战/握手认证协议)和 EAP(Extensible Authentication Protocol，可扩展认证协议)、UNIX 的登录操作(UNIX Login)等。

4) 数据包重传机制

RADIUS 采用 UDP 协议的原因有两点：一是客户端和 RADIUS 服务器大多在同一个

局域网中，使用 UDP 更加快捷方便；二是简化了服务端的实现。但是 UDP 协议存在丢包现象，所以 RADIUS 协议通过数据包重传机制解决 UDP 数据包丢失问题。

如果客户端在发出请求（接入请求、计费请求等）后没有收到响应信息，会多次重传请求，如果多次重传后仍然收不到响应，那么就认为 RADIUS 服务器已经关机。这时，客户端会向备用的 RADIUS 服务器发送请求。

5）重放攻击防范

为防止非法用户的重放攻击，如果在一个很短的时间片段内出现一个具有相同的客户源 IP 地址、源 UDP 端口号和标识符的请求，RADIUS 服务器将会认为这是一个重复请求，直接将其丢弃，不做任何处理。

5. RADIUS 协议的优势

RADIUS 协议简单明确，扩展性好，因此得到了广泛应用。该协议具有以下特点：

- 采用通用的客户/服务器结构组网。NAS 作为 RADIUS 的客户端负责将用户信息传递给指定的 RADIUS 服务器，然后处理 RADIUS 服务器的返回结果。RADIUS 服务器负责接收用户的连接请求，对用户进行认证，向客户端返回用户配置信息。
- 采用共享密钥保证网络传输安全性。客户端与 RADIUS 服务器之间的交互是通过共享密钥来进行相互认证的，以减少在不安全的网络中用户密码被侦听到的可能性。
- 具有良好的可扩展性。RADIUS 是一种可扩展的协议，所有的交互报文由多个不同长度的 ALV（Attribute-Length-Value，属性-长度-值）三元组组成，新增加属性和属性值不会破坏协议的原有实现。因此 RADIUS 协议也支持设备厂商扩充厂家专有属性。
- 协议认证机制灵活。RADIUS 协议认证机制灵活，支持多种认证用户的方式。如果用户提供了用户名和用户密码的明文，RADIUS 协议能够支持 PAP、CHAP、UNIX login 等多种认证方式。

RADIUS 协议简单明确，扩展性强，因此得到了广泛应用。在普通电话拨号上网、ADSL 拨号上网、社区宽带上网、VPDN 业务、移动电话预付费等业务中都能见到 RADIUS 的身影。

6. RADIUS 协议存在的问题

RADIUS 协议具有开放性、可扩展性、灵活性等优点，并且可以和在其他 AAA 安全协议（如 TACACS＋、Kerberos 等）共用。但是，随着网络技术的不断发展（例如移动 IP、NGN、3G 等），RADIUS 协议存在以下问题。

1）多协议支持

RADIUS 只支持 IP 协议，不支持 ARA（AppleTalk Remote Access，AppleTalk 远程访问）、NBFCP（NetBIOS Frame Control Protocol，网络基本输入输出系统帧控制协议）、IPX、X. 25 PAD connections（X. 25 PAD 连接）和 NASI（异步服务接口）等协议。

2）安全性

RADIUS 协议中，对用户密码属性采取的算法为 User-Password＝Password（不足 16 位填 0）XOR MD5（公用密钥＋请求认证），即用户密码是由原始的用户密码和公用密钥与

请求认证的 MD5 值的异或来表示的。针对这种算法，破坏者可以通过对大量截获的数据进行分析从而猜测用户密码，存在安全隐患。

RADIUS 协议采用的是共享密钥，而且用户密码以明文的方式存放于数据库中，所以系统内部的安全破坏（共享密钥的泄露、管理员的泄密）将会造成整个 AAA 功能的失效。另外，RADIUS 在认证或计费需要通过代理链的情况下无法提供端到端的安全性。

RADIUS 协议并不要求支持 IPSec 和 TLS，没有提供统一的传输层面上的安全。

3）可扩展性

当用户越来越多时，由于 RADIUS 协议中没有中继器和重定向器，所以只能不断增加新的 AAA 服务器。如果能够很好地支持中继、代理和重定向器，就可以把用户分组，把系统管理的能力分散到每个组，也能对来自不同组的请求加以集中处理，并转发到合适的目标，同时还能很好地实现负载均衡。

4）故障切换

RADIUS 中没有明确定义故障转移和故障恢复机制。

3.4　单点登录

所谓单点登录（SSO）是指在多个应用系统中，用户只需登录一次即可访问所有相互信任的应用系统，而不需要再进行额外的身份认证。IBM 公司对其有一个形象的解释："单点登录，全网漫游"。实施单点登录是目前流行的企业信息系统集成的重要组成部分，具有以下优点：

- 提高了用户工作效率。用户在不同系统中进行登录所耗费的时间减少了。由于用户不需要记忆多组账号和口令，也降低了用户登录出错的可能性。
- 方便了系统管理员对用户的管理。大多数单点登录系统采取对用户身份信息的集中存储，便于系统管理员增加、删除用户和修改用户权限。
- 增强了网络安全性。用户每使用一次身份凭证，就增加了一次凭证泄露和被截获的危险。当用户为了防止遗忘而将用户名、口令等记录下来时，就更增加了系统的安全隐患。

3.4.1　单点登录基本原理

单点登录的实质就是安全上下文（security context）或凭证（credential）在多个应用系统之间的传递或共享。假设有 3 个应用系统 A、B 和 C，使用单点登录后，用户经过一次身份验证就可以访问这 3 个授权的应用系统，登录流程如图 3.22 所示。

（1）当用户第一次访问应用系统（例如应用系统 A）时，由于尚未登录，会被引导到认证系统进行登录认证。

（2）根据用户提供的登录信息，认证系统进行身份校验，如果通过校验，则生成并返还给用户一个统一的认证凭据——票据；然后从认证系统跳转到 A 系统，用户成功访问 A 系统。

（3）用户再访问别的应用系统（例如应用系统 B 或 C）时带上这个票据，作为自己的身

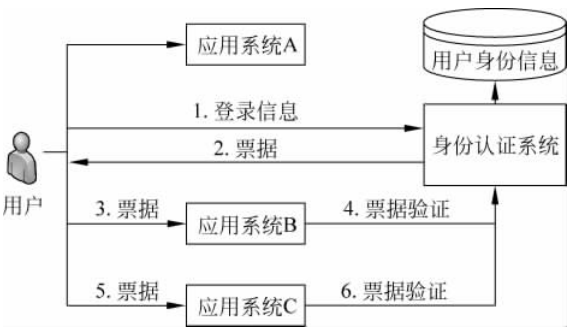

图 3.22 单点登录下用户登录流程

份凭据。

(4) 应用系统接收到请求后,把票据送到认证系统进行验证。如果通过验证,用户不用再次登录就可以访问应用系统 B 或 C 了。

票据在整个系统中是唯一的,绑定了时间戳和一些用户属性,用户无法通过伪造或交换票据来非法侵入系统。系统可以通过属性实现对用户访问的个性化控制。

从图 3.22 的流程可以看出,要实现单点登录,需要以下主要功能:

- 统一认证系统。所有应用系统共享一个身份认证系统是单点登录的前提之一。
- 识别票据。所有应用系统能够识别和提取票据信息,认证系统应该对票据进行校验,判断其有效性。
- 识别登录用户。应用系统能够能自动判断当前用户是否登录过,从而实现单点登录的功能。

上面的功能只是一个非常简单的单点登录架构,在实际应用中有着更加复杂的结构。有两点需要指出:

- 单一的用户信息数据库并不是必需的。有许多系统不能将所有的用户信息都集中存储,应该允许用户信息放置在不同的存储中。只要认证系统统一,票据的产生和校验统一,无论用户信息存储在什么地方,都能实现单点登录。
- 统一的认证系统并不是说只有单个认证服务器。整个系统可以存在多个认证服务器,这些服务器甚至可以是不同的产品。认证服务器之间通过标准的通信协议,例如 SAML(Security Assertion Markup Language),互换认证信息,从而实现更高级别的单点登录。

3.4.2 单点登录系统实现模型

实现单点登录的技术和模型主要有以下几种。

1. 基于经纪人(Broker-based)的 SSO 模型

在此模型中,有一个专门的服务器集中进行身份认证和用户账户管理,它负责给提出请求的用户发放身份标识,是一个公共和独立的“第三方”,可以形象地称其为“经纪人”。

如图 3.23 所示,该模型主要由 3 部分组成:支持认证服务的客户端、认证服务器和支持认证服务的应用系统。其工作流程如下:

(1) 客户端在访问系统资源之前,首先与认证服务器进行身份验证,获取电子身份标

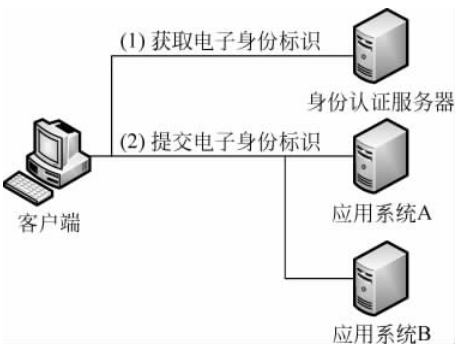

图 3.23　基于经纪人的 SSO 模型

识，为提高系统的安全性可以采用双向认证方式。

(2) 客户端凭借该身份标识访问各应用系统，实现单点登录。如果电子身份标识非法或者过期，应用系统应拒绝用户的访问。

基于本章前面介绍的 Kerberos 协议实现单点登录是此模型的典型应用。其他的还有 Sesame 和 Kryptoknight，Sesame(Secure European System for Application in Multivendor Environment)被认为是欧洲版本的 Kerberos，而 KryptoKnight 是 IBM 公司的一种类似于 Kerberos 的鉴别和密钥分配系统。

这种模型的特点如下：

- 从可实施性角度来看，该模型需要对现有应用系统进行改造，使其适应单点登录的认证机制，而改造旧系统的工作量通常较大，实施起来比较困难。
- 从可管理性角度来看，该模型对用户身份、权限、密钥等相关认证信息进行集中存储，易于进行管理和信息维护。但是，如果认证服务器失效，则所有的应用系统和用户都会受到影响，通常采用主/备认证服务器来提高系统的可靠性。
- 从安全性角度来看，实际的安全水平取决于所采用的认证协议的安全特性和系统工作机制。例如，Kerberos 中的认证仅基于口令，这就使系统容易受到口令猜测的攻击。
- 从可使用性角度来看，通过身份验证的客户端将持认证服务器返回的身份标识去访问应用系统，而不再与认证服务器打交道，减轻了认证服务器的工作负担，便于系统的扩展，也适用于大规模用户的环境。由于所有用户的登录信息都被系统接管，所以用户每次登录都要提供已经注册的账号和口令，匿名用户无法登录。

2. 基于代理(agent-based)的 SSO 模型

这是一种软件实现方式，如图 3.24 所示。在此模型中，被称为“代理”的程序可以运行在客户端或者服务器上，是客户端与应用系统之间的通信中介。若代理部署在客户端，它能装载获得账号/口令列表，自动替用户完成登录过程；若代理部署在应用系统服务器端，它就是服务器的认证系统和客户端认证方法之间的“翻译”。它可以使用口令表或加密密钥自动完成用户认证，从而免除用户进行认证的负担。

一个典型的基于代理模型的单点登录解决方案是 SSH。SSH 是目前较可靠、专为远程登录会话和其他网络服务提供安全性的协议，由客户端和服务器端的软件组成。

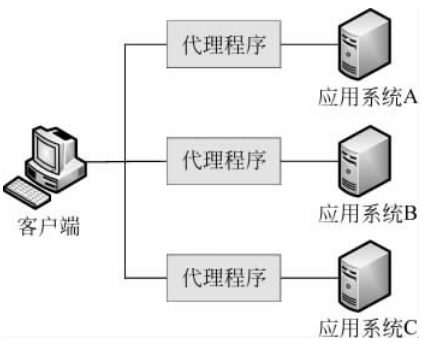

图 3.24 基于代理的 SSO 模型

服务器端是一个守护进程(daemon),在后台运行并响应来自客户端的连接请求,一般包括公共密钥认证、密钥交换、对称密钥加密和非安全连接。

客户端包含 ssh 程序以及像 scp(远程复制)、slogin(远程登录)、sftp(安全文件传输)等其他应用程序。SSH 的用户可以使用包括 RSA 算法等不同的认证方法。当使用 RSA 认证时,代理程序可以被用于单点登录。如果终端的代理程序有新的子连接产生,则继承原有连接的认证。利用 SSH 协议可以把所有传输的数据进行加密,有效防止远程管理过程中的信息泄露,从而避免 DNS 和 IP 欺骗等攻击。另外,使用 SSH 传输的数据是经过压缩的,可以加快数据传输的速度。

这种模型的特点如下:

- 从可实施性角度而言,该模型移植相对容易和灵活,但代理程序需要实现与原有应用系统的交互,即每个运行在主机(客户端或服务器)上的代理程序都要兼容现有的系统,增加了开发量,不具有良好的通用性。另外,它不适合跨域单点登录的实施。
- 从可管理性角度而言,每个应用系统都有各自的认证模块,用户身份信息是分散管理的,增加了管理难度,而且对各个代理的身份信息和权限也需要进行管理和设置。
- 从安全性角度而言,该模型要求用户的登录凭证在本地存储,增加了口令泄露的危险。采用有加密技术的认证协议可以保证代理程序的通信安全,但要保证代理软件本身的安全性。
- 从可使用性角度而言,该模型只要配置好代理软件,用户对应用系统的访问是透明的,使用方便。

3. 基于网关(gateway-based)的 SSO 模型

在此模型中,所有的客户端都与网关相连,网关再与各种应用服务器进行连接,所有的服务资源都放在被网关隔离的受信网段里。用户通过网关进行认证后获得访问服务的授权。

如图 3.25 所示,网关是通往所有服务资源必须经过的一道“门”,它可以是防火墙,也可以是专门用于通信加/解密的服务器。

基于网关的单点登录系统模型工作方式如下:

- 客户端与网关进行双向身份验证,即客户端要向网关证明自己是合法用户,同时网关也要向客户端证明自己是值得信赖的网关。

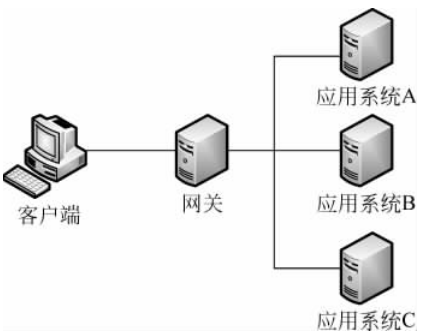

图 3.25　基于网关的 SSO 模型

- 客户端提出自己访问资源的请求，网关对用户进行认证，如果用户通过认证，网关则会授权用户使用对应的服务。由于在网关后的所有服务资源处在一个可被信赖的网络中，如果在网关后的服务能够通过 IP 地址进行识别，并在网关上建立一个基于 IP 的规则，而这个规则如果与在网关上的用户数据库相结合，网关就可以被用于单点登录。

基于网关模型与基于经纪人模型看起来类似，但两者的概念是有区别的。与经纪人模型不同的是，在用户登录时，网关可以记录客户端的身份，而不需要冗余的验证。因为网关控制着所有进入应用服务器的通道，可以监视和改变数据流，因此当用户想要进入时，它可以置换进入后的认证信息，把它传送到服务器，这样既能进行合适的访问控制，应用服务器自身又不需要改变。

这种模型的特点如下：

- 从可实施性角度而言，该模型对应用系统基本不做任何改变，客户端也不需要作太大变动，只要配置它们与网关相互认证的模块即可，实施也较为简单、快速。但是，在实施中对已有的网络环境要求比较严格，所以其应用范围并不广泛。
- 从可管理性角度而言，该模型中所有客户机通过网关来访问资源，可以对用户信息进行集中管理，减轻了网络管理负担。如果使用多个网关以克服瓶颈效应，那么这些网关中的用户数据要实现自动同步。
- 从安全性角度而言，该模型中网关的安全性至关重要，可以采取独立的防火墙来保护网关。
- 从可使用性角度而言，该模型的网关作为一个中心组件，它的性能会影响整个系统的效率，而且不适用于跨域的单点登录系统。

4. 基于令牌(token-based)的 SSO 模型

此模型典型的应用是由 RSA 公司提出的一个称为 SecurID 的解决方案。SecurID 采用双因子认证。第一个因子是用户身份识别码(PIN)，这是一串保密的数字，可由系统管理员定制。第二个因子是 SecurID Token，这是一个小型数字发生器，它每隔一段时间产生新的数字。这个发生器的时钟与网络环境中提供身份鉴别的服务器(ACE)保持同步，并且与 ACE 的用户数据库保持映射。“PIN＋同步时钟数字”就是用户的登录代码。

在基于令牌的 SSO 方案中也有一个称为 WebID 的模块。在 Web 服务器上安装一个

ACE 服务器的代理程序，用来接收 SecurID。当访问第一个需要认证的 URL 时，WebID 会使软件产生并加密一个标识，这个标识将在访问其他资源时被用到，从而实现单点登录功能。

这种模型的特点如下：

- 从可实施性角度而言，该模型需要增加新的组件，实施范围较狭窄。
- 从可管理性角度而言，由于该模型需要在系统上增加一些新的组件，因此增加了管理员的管理负担。
- 从安全性角度而言，基于令牌模型的最大特点就是它为用户产生基于时间间隔的一次性口令，增强了系统的安全性。
- 从可使用性角度而言，该模型需要额外的硬件和软件，用户掌握起来可能有困难。

从以上对 4 种主要的单点登录模型的介绍和评估可以看出，这些实现方案各有优缺点，所以在具体实施时要结合应用环境和各项安全技术进行综合考虑和设计。例如，将基于经理人模型和基于代理模型进行综合，如图 3.26 所示。

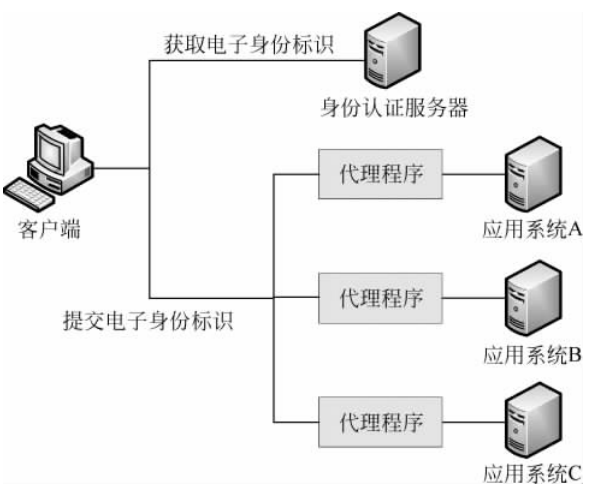

图 3.26 基于代理和经纪人的 SSO 模型

此方案比较适合大多数的应用环境，它一方面可以利用基于经纪人模型的集中管理机制，对用户进行统一的身份认证管理，另一方面又可以利用基于代理模型的灵活性，减少对原有应用系统的改造。

3.5 本章小结

本章首先通过几个典型案例引入了网络身份认证的概念和作用，接着列举了 3 种常用网络身份认证技术，即口令认证、IC 卡认证和基于生物特征识别的认证。结合密码技术介绍了对称密码认证和非对称密码认证，分析了 Kerberos 协议和 RADIUS 协议的工作过程和原理，描述了当前在电子商务和电子政务等领域得到广泛应用的 PKI 体系。最后，介绍了单点登录系统，它能简化服务之间的安全认证，提高服务之间的合作效率，已经成为系统

设计的基本功能之一。

3.6　本 章 习 题

1. 能够用于身份认证的人体生物特征有哪些？请举例说明。

2. PKI 的核心服务有哪些？

3. PKI 的认证服务有哪些优点？

4. PKI 有哪些组成部分，它们之间存在哪些关系？

5. PKI 系统是如何实现认证、保密、不可否认性的？

6. 在 PKI 中如何获取对方的证书和相关信息？

7. PKI 中实现证书存取库的方法有哪些？

8. 采用支持 LDAP 的目录服务器构造一个证书存取库。

9. SSO 的作用是什么？SSO 有哪些模型？

10. 在证书注册服务器上注册一个个人证书包括哪些步骤？试在安全网站上申请免费的个人证书。

11. 简述邮件加密软件 PGP 的加密体制和密钥管理策略，并用 PGP 实现对文件和邮件的加密传输。