第5章 chapter 5

认证技术

加密和认证是现代密码学的两大分支。加密的目的是防止敌方获取机密信息；认证的目的则是防止敌方欺骗、伪造、篡改、抵赖等形式的主动攻击。

认证(authentication)也称鉴别，是验证通信对象是原定者而不是冒名顶替者(身份认证)，或者确认收到的消息是希望的而不是伪造的或被篡改过的(消息认证)。认证技术包括身份认证和消息认证两大类。身份认证用于鉴别用户或实体的身份，而消息认证用于保证通信双方收到的信息的真实性和完整性。

认证技术的实现通常需要借助于加密和数字签名等密码学的技术。实际上，数字签名本身也是一种认证技术，它可用来鉴别消息的来源。

5.1 消息认证

消息认证是一个过程，用来验证接收消息的真实性(的确是由它所声称的实体发来的)和完整性(未被篡改、插入、删除)，同时还可用来验证消息的顺序性和时间性(未重排、重放、延迟)。

实现消息认证的手段可分为4类：①利用对称密码体制实现的消息认证；②利用公钥密码体制实现的消息认证；③利用散列函数实现的消息认证；④利用消息认证码实现的消息认证。

5.1.1 利用对称密码体制实现的消息认证

利用对称密码体制实现消息认证时，发送方A和接收方B事先共享一个密钥k。A用密钥k对消息m加密后通过公开信道传送给B。B接收到密文消息后，通过是否能用密钥k将其恢复成合法明文来判断消息是否来自A以及信息是否完整。利用对称密码体制实现消息认证如图5.1所示。

这种方法要求接收方有某种方法能判定解密后的明文是否合法，因此在处理中，可以规定合法的明文只能是属于在可能位模式上有微小差异的一个小子集，这使得任何伪造密文解密恢复出来后能成为合法明文的概率非常小。

在实际应用中，这是很容易实现的，可以假定明文是有意义的语句，而不是杂乱无章

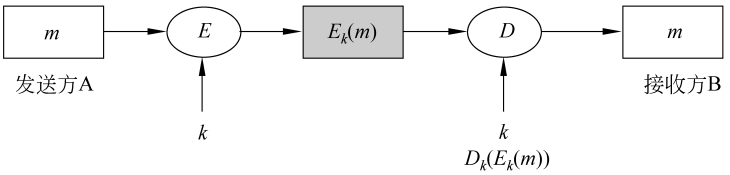

图 5.1 利用对称密码体制实现消息认证

的字符串。例如,将一个有意义的明文语句加密(无论使用什么算法)后,它都会以极大的概率变成一段杂乱无章的字符串,而几乎没有可能变成另一个有意义的语句。因此,如果发送方不知道密钥,用不正确的密钥(k')对明文加密,接收方收到后用正确的密钥 k 对密文解密,就相当于对密文再加密了一次,这样得到的是两次加密后的密文,有极大的概率仍然会是一段杂乱无章的字符串。所以,当接收方解密后发现明文是有意义的语句时,即使他不知道明文到底是什么内容,也可以以极大概率相信密文是发送方用正确的密钥加密得到的。

利用对称密码体制实现消息认证有如下几个特点:

(1) 能提供认证功能。可确认消息只可能来自 A,传输途中未被更改。

(2) 能提供机密性。因为只有 A 和 B 知道密钥 k。

(3) 不能提供数字签名功能。接收方可以伪造消息,发送方可以抵赖消息的发送。

提示:认证双方共享一个秘密就可以相互进行认证,这是最简单也是最常用的认证机制。例如,在现实生活中,如果两人知道某个共同的秘密(并且只有他们知道),就能依靠这个秘密进行相互认证。虽然该机制的原理很简单,但实现起来却要解决诸多问题,例如,如何让认证双方能够共享一个秘密,如何保证该秘密在传输过程中不会被他人窃取或利用,等等。

5.1.2 利用公钥密码体制实现的消息认证

1. 实现消息认证

如图 5.2 所示,在利用公钥密码体制实现消息认证时,发送方 A 用自己的私钥 SK_A 对消息进行加密,再通过公开信道传送给接收方 B;接收方 B 用 A 的公钥 PK_A 对得到的消息进行解密并完成鉴别。

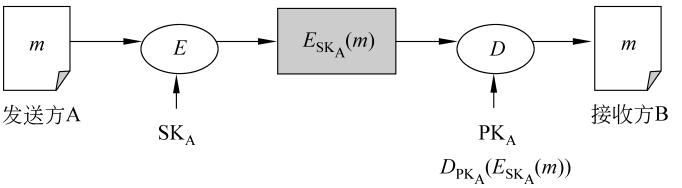

图 5.2 利用公钥密码体制实现消息认证

因为只有发送方 A 才能产生用公钥 PK_A 可解密的密文,所以消息一定来自拥有私钥 SK_A 的发送方 A。这种机制也要求明文具有某种内部结构,使接收方能够确定得到的

明文是正确的。

这种方法的特点是能提供消息认证和数字签名功能，但不能提供机密性，因为任何人都能用 A 的公钥将密文解密并查看消息。

2. 实现消息认证和保密性

如图 5.3 所示，当利用公钥密码体制实现消息认证和保密性时，发送方 A 用自己的私钥 SK_A 对消息进行加密消息认证（数字签名）之后，再用接收方 B 的公钥 PK_B 进行加密，从而实现机密性。这种方法能提供消息认证、数字签名功能和机密性。其缺点是一次完整的通信需要执行公钥算法的加密、解密操作各两次。

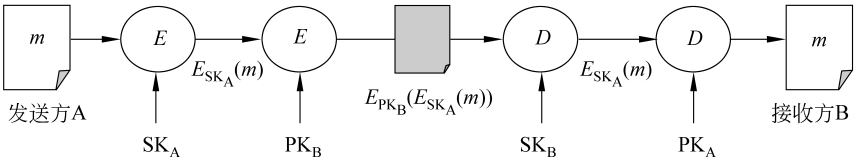

图 5.3　利用公钥密码体制实现消息认证和保密性

提示：通常情况下，都是先对消息进行签名再加密，因为被签名的消息应该是能够理解的。如果将消息加密之后再签名，则不符合常理，因为人们一般不会对一个看不懂的文件签名。当然，上述原则也不是绝对的，有时候也需要先加密再传送给别人签名，即盲签名。

5.1.3　利用散列函数实现的消息认证

散列函数具有以下特点：①输入是可变长度的消息 m，输出是固定长度的散列值（即消息摘要）；②计算简单，不需要使用密钥，具有强抗碰撞性。散列值只是输入消息的函数，只要输入消息有任何改变，就会输出不同的散列值，因此散列函数常常用于实现消息认证。

利用散列函数实现的消息认证有如下几种方案：

（1）用对称密码体制加密消息及其散列值，即 A→B：$E_k(m \parallel H(m))$，如图 5.4 所示。由于只有发送方 A 和接收方 B 共享密钥 k，因此通过对 $H(m)$ 的比较鉴别可以确定消息一定来自 A，并且未被修改过。散列值在方案中提供用于鉴别的冗余信息，同时 $H(m)$ 受到加密的保护，这样，该方案与用对称密钥直接加密消息相比，不要求消息具有一定的格式。该方案可提供机密性和消息认证，但不能提供数字签名。

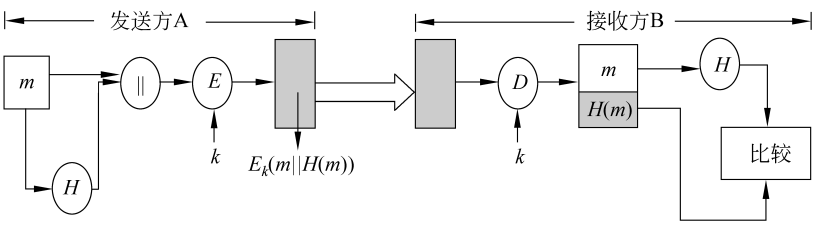

图 5.4　利用散列函数实现消息认证（方案 1）

(2) 用对称密码体制只对消息的散列值进行加密,并将散列值附在明文后,即 A→B: $m \| E_k(H(m))$,如图 5.5 所示。在该方案中消息以明文形式传递,因此不能提供机密性,但接收方可以计算 m 的散列值并与 $H(m)$比较,如果相同,就可以确定消息一定来自 A,并且消息 M 没有被篡改。该方法适用于对消息提供完整性保护而不要求保密性的场合,有助于减小处理代价。

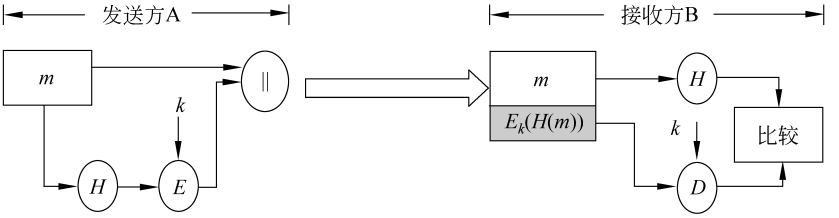

图 5.5 利用散列函数实现消息认证(方案 2)

(3) 用公钥密码体制的私钥对散列值进行加密,即 A→B: $m \| E_{kR_A}(H(m))$,如图 5.6 所示。该方案由于使用了发送方的私钥对 $H(m)$进行加密运算(实现了数字签名),因此可提供消息认证和数字签名。

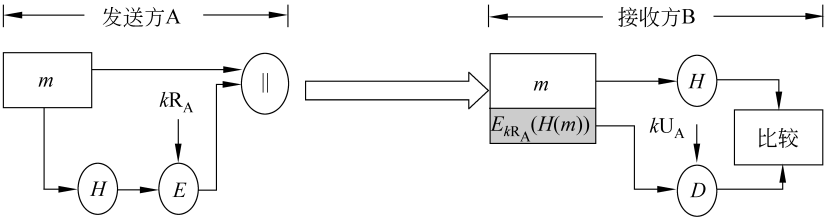

图 5.6 利用散列函数实现消息认证(方案 3)

(4) 结合使用公钥密码体制和对称密码体制,用发送方的私钥对散列值进行数字签名,然后用对称密钥加密消息 m 和签名的混合体,即 A→B: $E_k(m \| E_{kR_A}(H(m)))$,如图 5.7 所示。因此该方案既能提供消息认证和数字签名,又能提供机密性,在实际应用中较为常见。

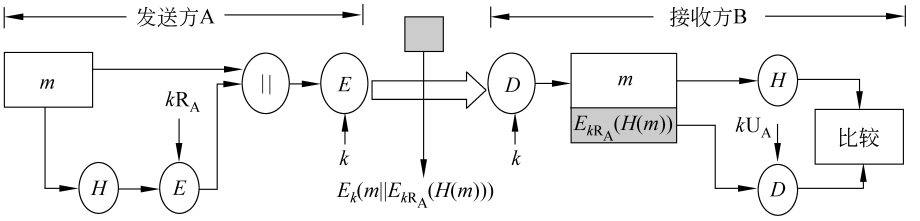

图 5.7 利用散列函数实现消息认证(方案 4)

(5) 使用散列函数,但不使用加密算法。为了实现消息认证,要求发送方 A 和接收方 B 共享一个秘密信息 s,发送方生成消息 m 和秘密信息 s 的散列值,然后与消息 m 一起发送给对方,即 A→B: $m \| H(m \| s)$,如图 5.8 所示。接收方 B 按照与发送方相同的

处理方式生成消息m和秘密信息s的散列值，对两者进行比较，从而实现消息认证。该方案的特点是：秘密信息s并不参与传递，因此可保证攻击者无法伪造。该方案又可被看成利用消息认证码实现消息认证，因为$H(m \| s)$可被看作m的消息认证码。

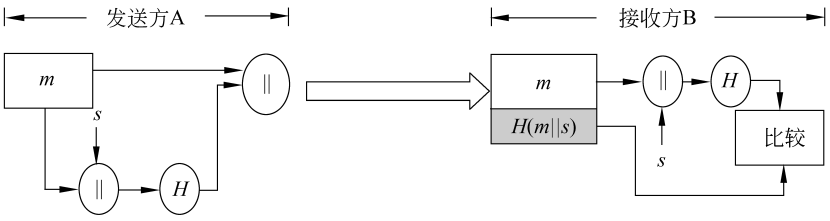

图 5.8　利用散列函数实现消息认证（方案 5）

（6）在方案（5）的基础上，使用对称密码体制对消息m和生成的散列值进行保护，即A→B：$E_k(m \| H(m \| s))$，如图 5.9 所示。这样，该方案除了能提供消息认证外，还能提供保密性。

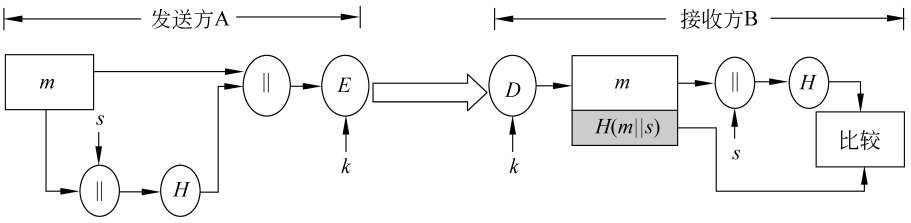

图 5.9　利用散列函数实现消息认证（方案 6）

5.1.4　利用消息认证码实现的消息认证

消息认证码（MAC）是用于提供数据原发认证和数据完整性保证的密码校验值。MAC 是消息被一个由密钥控制的公开散列函数作用后产生的固定长度的数值，用作认证符，此时需要通信双方 A 和 B 共享一密钥k，它由如下形式的函数产生：

$$\mathrm{MAC}=H_k(m)$$

其中，m是一个变长的消息，k是收发双方共享的密钥，$H_k(\cdot)$是密钥k控制下的公开散列函数。MAC 需要使用密钥k，这类似于加密，但两者的区别是产生 MAC 的函数是不可逆的，因为它使用带密钥的散列函数作为$H_k(\cdot)$来实现 MAC。另外，由于收发双方使用的是相同的密钥，因此单纯使用 MAC 是无法提供数字签名的。

对称密码体制和公钥密码体制都可以提供消息认证，为什么还要使用单独的 MAC 认证呢？

这是因为保密性和真实性是不同的概念。首先，从根本上讲，信息加密提供的是保密性而非真实性，而且加密运算的代价很大，公钥算法的代价更大。其次，认证函数与加密函数的分离有利于增强功能的灵活性，可以把加密和认证功能独立地实现在通信的不同传输层次。最后，某些信息只需要真实性，不需要保密性。例如，广播的信息量大，难

以实现加密；政府的公告等信息只需要保证真实性。因此，在大多数场合 MAC 更适合用来专门提供消息认证功能。

MAC 的基本用法有 3 种。

设 A 要发送给 B 的消息是 m，A 首先计算 MAC$=H_k(m)$，然后向 B 发送 $m'=m \parallel$ MAC。B 收到后进行与 A 相同的计算，求得新的 MAC′，并与收到的 MAC 做比较，如图 5.10 所示。如果二者相等，由于只有 A 和 B 知道密钥 k，故可以得到以下结论：

(1) 接收方 B 收到的消息 m 未被篡改。

(2) 消息 M'确实来自发送方 A。

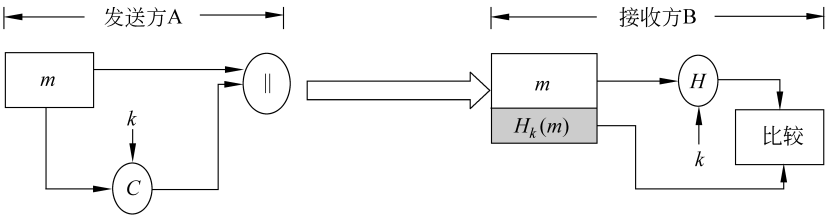

图 5.10 用 MAC 实现消息认证

图 5.10 中的方法只能提供消息认证，不能提供保密性。为了提供保密性，可以在生成 MAC 之前(图 5.11)或之后(图 5.12)使用加密机制。后两种方法生成的 MAC 或者基于明文，或者基于密文，因此相应的消息认证或者与明文有关，或者与密文有关。一般来说，基于明文生成 MAC 的方法在实际应用中更方便一些。

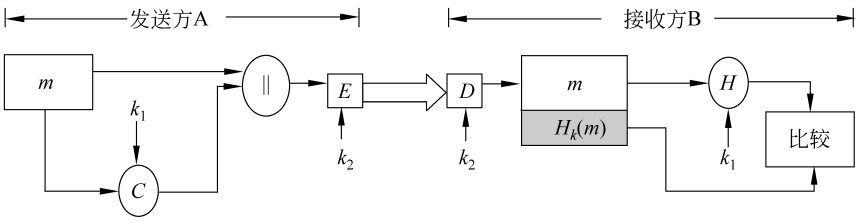

图 5.11 用 MAC 实现消息鉴别与保密性(与明文相关)

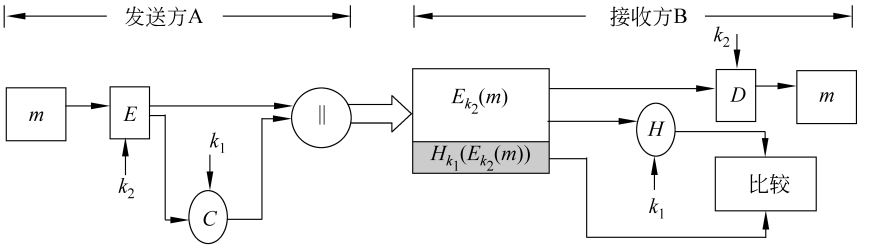

图 5.12 用 MAC 实现消息鉴别与保密性(与密文相关)

在实际应用中，对于电子商务安全等基于 Internet 的应用，一般采用公钥密码体制或散列函数进行消息认证；而对于物联网安全或移动支付等应用，则更多地采用对称密

码体制结合散列函数进行消息认证。这是由终端的计算和存储能力及网络带宽决定的。

5.1.5 数字签密

在信息安全服务中，为了能够同时保证消息的保密性、完整性、真实性和不可抵赖性等安全要素，传统的方法是对消息先签名后加密，这种方法的计算量和通信成本是加密和签名的代价之和，因此效率低下。为此，人们又提出了数字签密（digital signcryption）体制，即对消息同时进行签名和加密。

数字签密是1997年在美洲密码学会上由Y. Zheng提出的，它把传统的数字签名和公钥加密两个功能合并到一个步骤中完成。数字签密具有以下优点：

（1）签密在计算量和通信成本上都要低于传统的先签名后加密方法，例如，Y. Zheng提出的签密方案比基于离散对数问题的先签名后加密方法可节省58%的计算量和70%的通信成本。

（2）签密允许并行计算一些昂贵的密码操作。

（3）合理设计的签密方案较传统方案具有更高的安全水平。

（4）签密可以简化同时需要机密性和消息认证的密码协议的设计。

根据公钥认证方法的不同，数字签密体制可分为基于PKI的签密体制、基于身份的签密体制和无证书签密体制。

基于PKI的签密体制一般由3个算法组成：①密钥生成算法（keygen）；②签密算法（signcrypt）；③解签密算法（unsigncrypt）。这些算法必须满足签密体制的一致性约束。即，如果密文$\sigma=\text{Signcrypt}(m,\text{SK}_s,\text{PK}_r)$，那么明文$m=\text{Unsigncrypt}(\sigma,\text{PK}_s,\text{SK}_r)$。其中，$\text{SK}_s$和$\text{PK}_s$分别是发送方的私钥和公钥，$\text{SK}_r$和$\text{PK}_r$分别是接收方的私钥和公钥。

5.2 身份认证

身份认证是指证实用户的真实身份与其所声称的身份是否相符的过程。身份认证是所有安全通信的第一步，因为只有确信对方是谁，通信才有意义。身份认证的主要方式是基于秘密。通常，被认证者和认证者之间共享同一个秘密（如口令）；或者被认证者知道一个值，而认证者知道从这个值推出的值。

正确识别用户、客户机或服务器的身份是信息安全的重要保障之一。典型的例子是银行系统的自动取款机，用户可以从自动取款机中提取现金，但前提是银行首先要认证用户身份，否则恶意的假冒者会使银行或用户遭受损失。同样，对计算机系统的访问也必须进行身份认证，这不仅是网络安全的需要，也是社会管理的需要。

5.2.1 身份认证的依据

身份认证的依据可分为3类：

（1）用户知道的某种信息，如口令或某个秘密。

（2）用户拥有的某种物品，如身份证、银行卡、密钥盘、令牌、IP地址等。

(3) 用户具有的某种特征,如指纹、虹膜、DNA、脸型等。

这 3 类依据对应的认证方式各有利弊。第一类最简单,系统开销最小,但是安全性最低,这种方式在目前很多对安全性要求不高的网站上仍然最常用;第二类泄露秘密的可能性较小,安全性比第一类高,但是相对复杂;第三类的安全性最高,例如假冒一个人的指纹相当难,但这种方式需要购买昂贵的鉴别设备,并且只能对人进行认证,而 Internet 上更多的是需要对主机或程序进行认证。

有时候也把几类认证方式综合起来使用。例如,用户从自动取款机取款,必须拥有银行卡,还必须知道银行卡的口令,才能通过自动取款机的身份认证。这种使用两种依据的认证叫作双因素(two-factor)认证方式。

提示:5.1 节中介绍的很多消息认证方法也能用来实现身份认证,因为利用消息中包含的某些特殊的特征就能判断该消息一定是由某人发出的,因此可以证实消息发送方的身份。但这些方法安全性不高,因为攻击者截获消息后再转发给接收方就能进行冒充了。

5.2.2 身份认证系统的组成

身份认证系统一般由以下几部分组成:

(1) 示证者(Prover,P),又称声称者(claimant)。示证者提交一个实体的身份并声称他是那个实体。

(2) 验证者(Verifier,V)。验证者检验示证者提出的身份的正确性和合法性,决定是否满足其要求。

(3) 可信第三方(Trusted third Party,TP)。可信第三方参与调解纠纷,在安全相关活动中,它被双方实体信任。当然,有些简单的身份认证系统不需要可信第三方。

(4) 攻击者。攻击者可以窃听或伪装示证者,骗取验证者的信任。

身份认证系统的组成如图 5.13 所示。

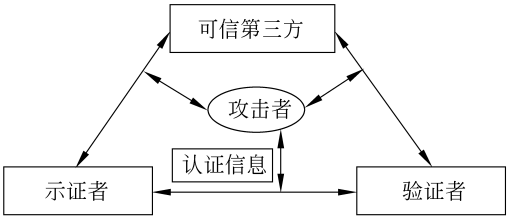

图 5.13 身份认证系统的组成

5.2.3 身份认证的分类

身份认证可分为单向认证和双向认证。单向认证是指通信双方中只有一方对另一方进行身份认证,而双向认证是指通信双方相互进行身份认证。

在单向认证中,一个实体充当示证者,另一个实体充当验证者。例如,一般的网站就采用单向认证,只有网站能验证用户的身份,而用户无法验证网站的真伪。

在双向认证中,每个实体同时充当示证者和验证者,互相进行身份认证。例如,在电

子商务活动中，双向认证能提供更高的安全性。双向认证可以在两个方向上使用相同或不同的认证机制。

身份认证还可分为非密码的认证机制和基于密码算法的认证机制。非密码的认证机制包括口令机制、一次性口令机制、挑战-应答机制、基于生物特征的机制等；基于密码算法的认证机制主要采用双方共享一个验证密钥等方法，与消息认证采用的方法类似。

5.3 口令机制

口令是目前使用最广泛的身份认证机制。从形式上看，口令是由字母、数字或特殊字符构成的字符串，只有被认证者知道。

提示：在日常生活中所说的银行卡密码、邮箱登录密码、保险柜密码等，准确地说应该叫口令，因为密码（密钥）是用来加密信息的，而口令是用来作为某种鉴别的秘密。

5.3.1 口令的基本工作原理

最简单的口令工作原理是：用户在注册时自己选择一个用户名和口令，或者系统为每个用户指定一个用户名和初始口令，用户可以定期改变口令，以保证安全性。口令以明文形式和用户名一起存放在服务器的用户数据库中。这种口令机制的工作过程如下：

第一步，系统提示用户输入用户名和口令。

认证时，应用程序向用户发送一个登录界面，提示用户输入用户名和口令（登录界面上通常使用“密码”），如图 5.14 所示。

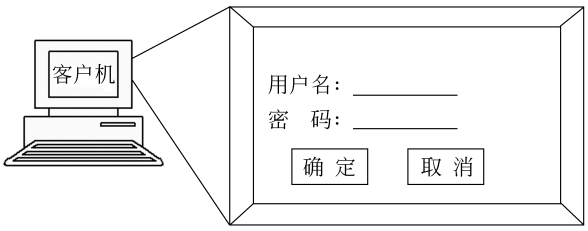

图 5.14　登录界面

第二步，用户输入用户名和口令，并单击“确定”按钮，使用户名和口令以明文形式传递到服务器上，如图 5.15 所示。

图 5.15　用户发送登录请求

第三步，服务器验证用户名和口令。

服务器中存储了用户数据库，通过该数据库检查这个用户名和口令是否存在并且匹

配，如图 5.16 所示。通常这是由用户鉴别程序完成的，该程序首先获取用户名和口令，在用户数据库中检查，然后返回鉴别结果（成功或失败）给服务器。

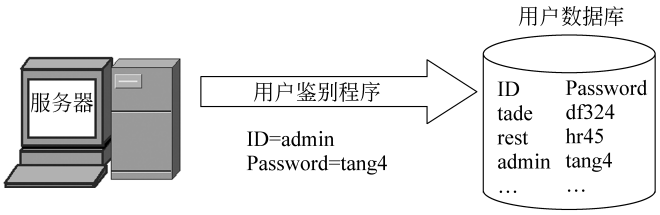

图 5.16 用户鉴别程序通过用户数据库检查用户名和口令

第四步，服务器通知用户。

根据检查结果，服务器向用户返回相应的界面。例如，如果用户鉴别成功，则服务器发送给用户一个菜单，列出用户可以进行的操作；如果用户鉴别不成功，服务器向用户发送一个错误信息界面。这里假设用户鉴别成功，如图 5.17 所示。

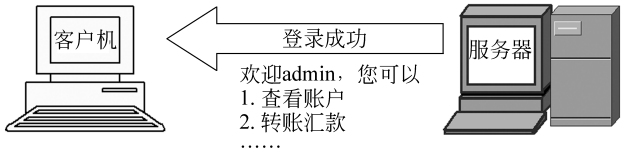

图 5.17 服务器向用户返回鉴别结果

5.3.2 对口令机制的改进


口令机制

5.3.1 节的口令方案可抽象成一个身份认证模型，如图 5.18 所示。该身份认证模型包括示证者和验证者，图 5.15 中的客户机是示证者，而拥有用户数据库的服务器是验证者。

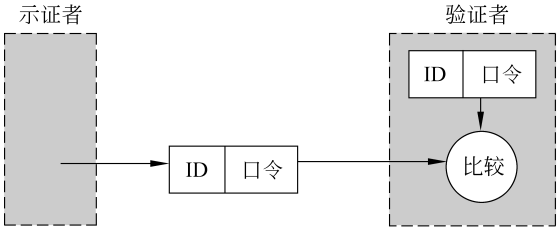

图 5.18 采用口令机制的身份认证模型

但是图 5.18 中的口令机制是很脆弱的，最严重的问题是口令可能遭受线路窃听、危及验证者攻击和重放攻击等。本节介绍前两种情况的应对措施。

1. 对付线路窃听的措施

如果攻击者对传输口令的通信线路进行窃听，就可能获得用户名和口令的明文，冒充合法用户进行登录。在目前的广播式网络中，通过抓包软件截获用户传输的认证信息