第19章 基本应用

19.1 身份认证

身份认证服务提供一种鉴别实体真伪的方法。由于数字证书在申请时身份已经得到发证机构确认，所以把数字证书用于应用系统对用户的身份认证是非常合适的。在应用中集成数字证书进行身份认证时，主要是验证用户是否拥有证书对应的私钥，即验证用户私钥对一段特定数据的签名是否正确。

例如，网上银行系统中，用户使用U盾（即USBKey）登录网银，U盾中存储了用户的证书和私钥，并通过密码对U盾进行保护。网上银行证书登录界面如图19-1所示。

图19-1 证书登录输入界面

界面输入元素含义分别为：

① 设备类型，表明采用的设备类型，网银支持多种类型的设备，如U盾、动态口令等；

② 用户ID，一个容易记的名称；

③ 数字证书，显示设备中证书对应的使用者名称；

④ 证书密码，设备保护密码；

⑤ 验证码，图片对应字符，防止重复提交。

当用户单击“提交”按钮时，登录功能主要做了两件事：

① 通过“证书密码”输入值验证用户是否有访问U盾的权限，此密码保护U盾不被非法访问；

② 在获取访问U盾的权限后，使用U盾中的私钥对服务端传递来的一段随机数据进行签名，然后把签名值和证书提交到服务端，服务端使用发送给客户端的随机数据，以及用户提交的证书对签名值进行验证。如果验证通过，表明用户拥有证书对应私钥及其使用权限；如果验证没有通过，表明用户没有证书对应私钥的使用权限。

即使用户通过了签名值验证阶段，也不一定能够通过身份认证。一般情况下，业务系

统还会对用户证书的有效性进行验证。有效性验证包括：证书是否过期、是否作废、是否为指定 CA 签发、密钥用途等。

网上银行用户的身份认证（证书登录）过程是典型的挑战应答模式，由服务端给出挑战值，即一段随机数据，客户端对挑战值进行应答，即对随机数据进行签名，然后提交应答（包括签名值、证书等）。

出于安全考虑，为了防止重复攻击，每次身份认证时，服务端都应该给出不同的挑战值，使挑战值尽可能随机化。

《卫生系统数字证书应用集成规范》中给出了数字证书登录的认证流程，如图 19-2 所示。

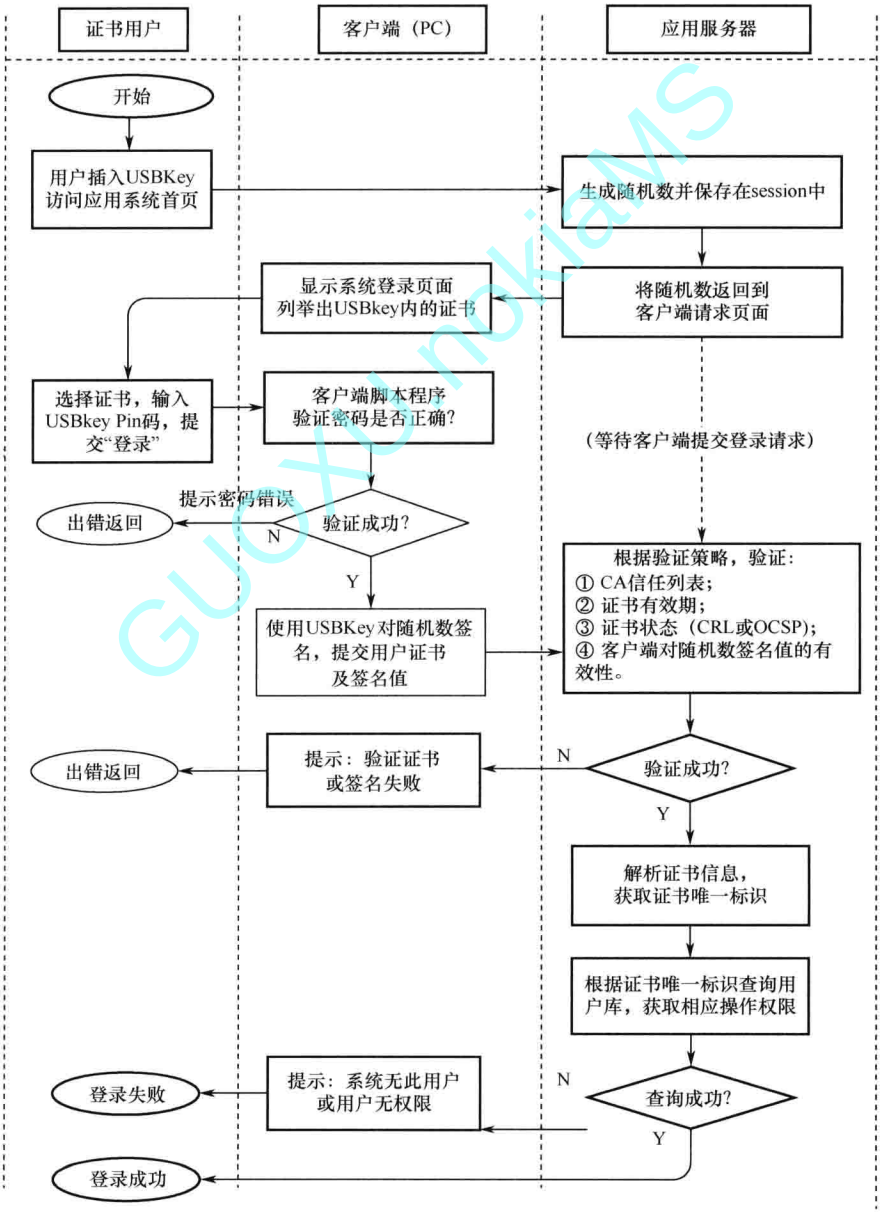

图 19-2　证书登录认证流程示意图

19.2 保密性

保密性是对数据进行加密的操作，使数据只能被授权的实体看到，它适用于以下几种场合。

（1）自己加密，自己解密

在这种情况下，实体出于保护自己数据的目的，使用自己的证书公钥对数据进行加密，也只有自己的私钥才能解开。

例如，用户有敏感信息需要保存，这些信息不希望被其他人看到，这样他就可以用自己的证书公钥对敏感信息加密，因私钥由自己掌控，其他人即使拿到加密后的数据，也无法打开。

（2）自己加密，他人解密

在这种情况下，用户出于保护要传递数据的目的，使用对方的证书公钥对数据进行加密，也只有对方的私钥才能解开。

例如，甲要向乙发送数据，而甲不希望除乙之外的其他人看到发送数据的内容，这时，甲就可以用乙的证书公钥加密数据，也只有乙能够用自己的私钥解开接收到的数据。平时所用的加密邮件，就属于此种类型应用。

由于使用公钥加密的效率较低，当需要加密的数据较多时，这种效率是不能接受的。为了解决这个问题，一般采用数字信封机制，即先产生一个对称密钥，使用对称密钥对数据进行加密，然后使用证书公钥对对称密钥加密，这样加密的对称密钥和加密后的数据形成了一个数字信封。解密时，进行反向操作，先用私钥解密对称密钥，然后用对称密钥解密加密的数据，得到数据原文。

构造数字信封时，一般使用 PKCS#7 中定义的数字信封格式，具体可参见第 5 章的内容。

下面以加密邮件说明数据的保密性。一封简单加密邮件包含如下信息：

```
Content-Type: application/pkcs7-mime; smime-type=enveloped-data;
name=smime.p7m
Content-Transfer-Encoding: base64
Content-Disposition: attachment; filename=smime.p7m
        rfvbnj756tbBghyHhHUujhJhjH77n8HHGT9HG4VQpfyF467GhIGfHfYT6
7n8HHGghyHhHUujhJh4VQpfyF467GhIGfHfYGTrfvbnjT6jH7756tbB9H
f8HHGTrfvhJhjH776tbB9HG4VQbnj7567GhIGfHfYT6ghyHhHUujpfyF4
0GhIGfHfQbnj756YT64V
```

邮件内容类型为 application/pkcs7-mime，smime-type=enveloped-data 表明是信封数据，加密后邮件正文采用 Base64 编码。

发送加密邮件时，加密操作在发送邮件时执行，捕获电子邮件未加密原文，并使用预期收件人所特有的信息（证书公钥）对邮件进行加密。加密的邮件替换了原始邮件，然后将邮件发送至收件人。图 19-3 显示了电子邮件加密过程。

由于此操作需要收件人的唯一信息（证书公钥），因此邮件加密提供了保密性。只有预期收件人具有执行解密操作所需的信息，从而确保了只有预期的收件人能够查看邮件。

当收件人打开加密邮件时，会对加密邮件执行解密操作。此时，将同时检索加密的邮件和收件人的唯一信息。然后，使用收件人的唯一信息对加密邮件执行解密操作。此操作

图 19-3　加密邮件过程

返回未加密的邮件，然后该邮件将显示给收件人。如果邮件在传送过程中发生过改变，解密操作将失败。图 19-4 显示了解密电子邮件过程的顺序。

图 19-4　解密电子邮件

19.3　完整性

完整性是指在传输、存储信息或数据的过程中，确保信息或数据不被未授权地篡改或在篡改后能够被迅速发现。这种方法实质是一种数据摘要过程，首先利用哈希函数计算数据哈希值，然后将数据及其哈希值一起发送到对方。对方收到数据后，重新利用哈希函数计算数据哈希值并与接收到的数据哈希值比对，进而判断数据是否被篡改。

由于哈希值在传输中可能被截取修改，需要对哈希值进行保护。一般通过数字签名的方式对哈希值进行保护，接收方对收到的数据的数字签名进行验证，如果数据原文被修改，则签名验证就不会通过。

在应用中，一般使用 PKCS#7 的签名数据结构组装数据和签名值。例如，发送签名邮件时，就是使用 PKCS#7 的签名数据结构封装消息，具体可参见第 5 章的内容。

下面以签名邮件说明数据的完整性。一封简单签名邮件包含如下信息。

```
Content-Type: application/pkcs7-mime; smime-type=signed-data;
name=smime.p7m
Content-Transfer-Encoding: base64
Content-Disposition: attachment; filename=smime.p7m

567GhIGfHfYT6ghyHhHUujpfyF4f8HHGTrfvhJhjH776tbB9HG4VQbnj7
77n8HHGT9HG4VQpfyF467GhIGfHfYT6rfvbnj756tbBghyHhHUujhJhjH
HUujhJh4VQpfyF467GhIGfHfYGTrfvbnjT6jH7756tbB9H7n8HHGghyHh
6YT64V0GhIGfHfQbnj75
```

内容类型为 application/pkcs7-mime，smime-type=signed-data 表明是签名数据，签名后邮件正文采用 Base64 编码。

发送签名邮件时，执行签名操作所需要的信息只能由发件人提供。在签名操作中，通过捕获电子邮件并对邮件执行签名操作来使用此信息。此操作产生实际的数字签名。然后，此签名将附加到电子邮件中，并随同邮件一起发送。图 19-5 显示了邮件签名过程的顺序。

图 19-5　发送签名邮件

由于此操作需要来自发件人的唯一信息（私钥），因此数字签名提供了身份验证和数据完整性功能。此唯一信息可以证明邮件只能来自该发件人。

当收件人打开经过数字签名的电子邮件时，系统会对数字签名执行验证过程，并会从邮件中检索邮件所包含的数字签名。还会检索原始邮件，然后执行签名验证操作，如果签名验证通过，则证明邮件确实来自所声称的那个发件人；如果签名验证不通过，则将邮件标记为无效。图 19-6 显示了邮件验证过程的顺序。

图 19-6　验证签名邮件

同时采用数字签名过程和数字签名验证过程可验证电子邮件发件人的身份，并确定已签名的邮件中数据的完整性。

19.4　抗抵赖性

抗抵赖提供一种防止实体对其行为进行抵赖的机制，它从技术上保证实体对其行为的认可。由于实体的各种行为只能发生在它被信任之后，所以可通过时间戳标记和数字签名来审计实体的各种行为。通过将实体的各种行为与时间和数字签名绑定在一起使实体无法抵赖其行为。

采用单纯的数字签名方式不能起到抗抵赖作用，因为签名说明签名者对数据或行为进行了确认，但不能说明他什么时间进行了确认，因此还必须进行可信的第三方进行事件发生时间的证明，即需要用到时间戳服务。

简单地说，时间戳服务是对一段数据进行时间标记，并对时间标记进行签名。在 RFC 3161 中对时间戳协议进行了说明。可信时间戳服务的本质是将用户的电子数据的哈希值和权威时间源绑定，在此基础上通过时间戳服务中心（TSA）进行数字签名，产生不可伪造的时间戳数据。时间戳协议在第 20 章有详细说明。

为了达到抗抵赖效果，对原始数据的签名包括两部分，一是实体对数据的签名，二是时间戳服务中心对数据的签名。这两部分签名分别使用了 PKCS#7 和 CMS 的签名数据结