第5章　信 任 认 证

在网络中，为了安全，发送和接收两方要互相信任才可以进行通信，交换信息。通信双方的信任是通过认证建立的。消息认证（message authentication）和身份认证（identity authentication）是建立信息系统安全信任体的两个核心环节，也是本章的核心内容。

消息认证也称为报文鉴别，是在数据机密性保护的基础上，对接受到的消息可信任度进行评估的机制。所评定的内容包括消息的完整性、真实性和不可抵赖性三个方面。

确保所接收到的报文具有完整性信息安全措施，是报文接收方检验报文是否可信的手段，涉及消息摘要（Message-Digest，DM）、消息认证码（Message Authentication Code，MAC）、数字签名（digital signature）三种技术。

身份认证是信息系统的第一道屏障，用于检验行为主体身份的可信任性。这种信任关系通常是通过单向——被访问者（系统）向访问者的授权建立的。认证的目的是检测访问者是否已经被授权，以控制哪些用户能够登录到系统（服务器）并获取系统资源。在介绍了信息系统安全的两个核心认证之后，本章还要介绍在去中心化环境和互联网中的信任认证机制。两个实体交互时往往需要进行双向认证，以确定双方的互相信任关系。

5.1　消 息 摘 要

5.1.1　数据的完整性保护与消息摘要

1. 数据的完整性

数据的完整性保护包括了对内容完整性、序列完整性和时间完整性的保护，是针对如下 3 种攻击采取的对策。

（1）内容篡改（content modification），包括对报文内容的插入、删除、改变等。

（2）序列篡改（sequence modification），包括对报文序列的插入、删除、错序等。

（3）时间篡改（timing modification），对报文进行延迟或回放。

2. 消息摘要

如图 5.1 所示，消息摘要是通过某种散列（hash）算法将任意长度的二进制数据映射为固定长度的、较小的二进制，这个小的二进制值称为哈希值（hash code）。

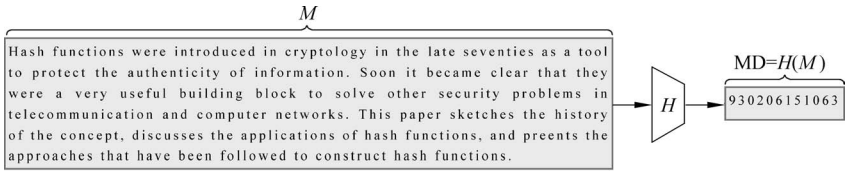


图 5.1　报文摘要的生成

哈希值的特点是唯一、定长、不可逆，即不管原始消息（报文）是什么样，所得到的摘要一定是唯一的、定长的，只要对原始消息进行一点修改，不管是内容的、序列的，还是时间的修改，用同样的算法，所得到的摘要一定不再相同，并且无法用摘要获得原消息。

这样，如图 5.2 所示，若消息发送者在发送消息 *M* 时，连同摘要 *H*(*M*)一起发送，则接收方就可以用同样的算法对原始消息再映射出一个摘要来，将两个摘要进行比对，就可以知道收到的消息是否经过了篡改。图 5.2 中，*H* 为散列函数，*K* 为对称密钥，*E* 为用 *K* 加密后的密文，*D* 为用 *K* 解密后的明文。

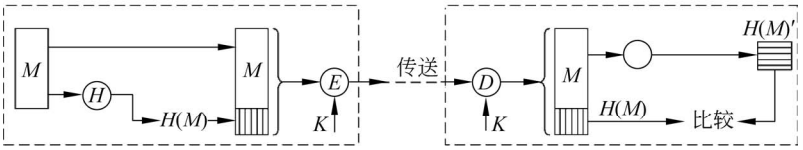


图 5.2　用消息摘要对消息的完整性进行认证

3. 哈希函数的特征

消息摘要的关键是使用了什么样的散列算法（函数）。作为哈希函数应具备以下性质。

（1）*H*()可应用于任意长度的输入数据块，产生固定长度的哈希值。

（2）对于每一个给定的输入数据 *M*，计算出它的哈希值 *h* = *H*(*M*)很容易。

（3）给定哈希值 *h*，倒推出输入数据 *M* 在计算上不可行，即单向性。

（4）对于给定的报文 *M* 和其哈希值 *h*，要找到另一个 *M* ′≠*M*，使 *H*(*M* ′)=*H*(*M*)极其困难，即弱抗碰撞（collision）性。

（5）找到任何满足 *H*(*M*)=*H*(*M* ′)且 *M*≠*M* ′的报文对(*M*, *M* ′)在计算上是不可行的，即强抗碰撞性。碰撞性是指对于两个不同的报文，如果它们的摘要值相同，则发生了碰撞。

（6）哈希函数并不提供机密性，并且它们不使用密钥生成摘要。

表 5.1 为目前常用无碰撞散列算法一览表。

表 5.1　目前常用无碰撞散列（Hash）算法

算法名称	输出大小（bit）	内部大小	区块大小	长度大小	字符尺寸	碰撞情形
HAVAL	256/224/192/160/128	256	1024	64	32	是
MD2	128	384	128	No	8	大多数
MD4	128	128	512	64	32	是
MD5	128	128	512	64	32	是
PANAMA	256	8736	256	否	32	是
RadioGatún	任意长度	58 字	3 字	否	1-64	否
RIPEMD	128	128	512	64	32	是
RIPEMD-128/256	128/256	128/256	512	64	32	否
RIPEMD-160/320	160/320	160/320	512	64	32	否
SHA-0	160	160	512	64	32	是
SHA-1	160	160	512	64	32	有缺陷
SHA-256/224	256/224	256	512	64	32	否
SHA-512/384	512/384	512	1024	128	64	否
Tiger（2）-192/160/128	192/160/128	192	512	64	64	否

5.1.2 报文摘要算法

1. 报文摘要算法及其应用

报文摘要算法（Message-Digest algorithm 5，MD5）是沿着 MD2、MD4 的轨迹进化形成的报文摘要算法的新版本。它的开发者是作为 RSA 算法设计者之一的 Ronald L.Riverst。MD5 不以任何假设和密码体制为基础，是一个直接构造出来的算法。它的主要特点如下。

（1）单向性。可以由报文生成报文摘要，但不能由报文摘要还原为报文。

（2）无碰撞性。不同的报文不会产生两个相同的报文摘要。

（3）运算速度快。应用比较普遍。

（4）MD5 主要应用在两个方面：防篡改鉴别和加密。

MD5 的典型应用是对一段报文产生一个 128b 的报文摘要。例如，在 UNIX 下有很多软件在下载的时候都有一个扩展名为 md5 的文件，在这个文件中通常只有一行文本，大致结构如下。

```
MD5（tanajiya.tar.gz）= 0ca175b9c0f726a831d895e269332461
```

这就是 tanajiya.tar.gz 文件的数字签名。如果在以后传播这个文件的过程中，无论文件的内容发生了任何形式的改变（包括人为修改或者下载过程中线路不稳定引起的传输错误等），只要对这个文件重新计算 MD5 值就会发现报文摘要不相同，由此可以确定得到的只是一个不正确的文件。如果再有一个第三方的鉴别机构，用 MD5 还可以防止文件作者的“抵赖”，这就是数字签名的应用。

在加密系统中，密码的保管非常重要。由于具有系统管理员权限的用户可以读密码文件，所以常规保存的密码文件对具有系统管理员权限的用户就无秘密可言。而 MD5 的单向性和无碰撞性使得用户密码可以用 MD5（或其他类似的算法）加密后存储。当用户登录的时候，系统把用户输入的密码计算成 MD5 值，然后再去和保存在文件系统中的 MD5 值进行比较，进而确定输入的密码是否正确。这样，系统就可以在不知道用户密码的明码的情况下确定用户登录系统的合法性，不但可以避免用户的密码被知道，而且还在一定程度上增加了密码被破解的难度。

2．MD5 算法的基本轮廓

（1）以 512b 分组来处理输入的报文。

（2）每一分组又被划分为 16 个 32b 子分组。

（3）经过了一系列的处理后，输出 4 个 32b 分组放在 4 个链接变量（chaining variable）或寄存器 *A*、*B*、*C*、*D* 中。

（4）将 4 个链接变量进行级联，生成一个 128b 的哈希值。

3. MD5 算法的完成步骤

（1）数据扩展。将报文按照图 5.3 的格式用 100…0 进行填充，并附加一个 64b 的原始报文长度字段，使得总长度为 512b 的整数倍。应当注意，填充是必需的，若报文长度为（512b 的整数倍–64b），则还需填充一个 512b 长度的填充字段。

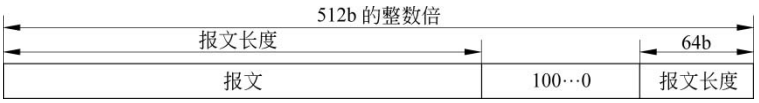


图 5.3　数据准备格式

（2）初始化 MD 缓冲区。使它们的十六进制初始值分别为

A = 0x01234567，B = 0x89abcdef，C = 0xfedcba98，D = 0x76543210

（3）报文切块。按照 512b 的长度，将报文分割成 N + 1 个分组：Y_0、Y_1、…、Y_N。每一分组又可以表示为 16 个 32b 的字。

（4）依次对各分组进行 H_{MD5} 压缩生成 MD5 值。如图 5.4 所示，从分组 Y_0 开始到最后一个分组 Y_N，依次进行 H_{MD5} 压缩运算。每个 H_{MD5} 有两个输入（一个 128b 的 CV_q 和一个 512b 的分组 Y_q）和一个 128b 输出；最开始的 128b 输入为 4 个 32b 的链接变量；以后每个 H_{MD5} 的输出作为下一个 128b 输入。最后一个 128b 输出即所求的 MD5 值。显然，这个过程可以用循环或递归实现。

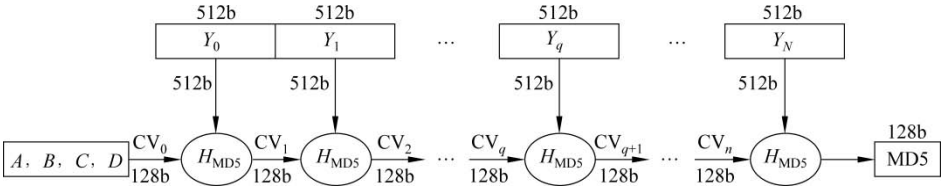


图 5.4　依次对各分组进行 H_{MD5} 压缩生成 MD5 值

H_{MD5} 算法是 MD5 的关键函数。它的计算由 4 轮处理组成，每一轮又包括了 16 个操作，总共 64 次操作，每一次操作要使用一个常数。关于它的细节这里不再介绍。需要说明的是，MD5 曾经被人们认为是无碰撞的。但是，在 2004 年 8 月 17 日在美国加州圣巴巴拉召开的国际密码学会议（Crypto’2004）上，我国山东大学教授王小云宣布她已经找到了使 MD5 产生碰撞的方法。但是，目前人们还没有找到 MD5 的合适的代替方案。

5.1.3　安全哈希算法

安全哈希算法（Secure Hash Algorithm，SHA）是由 NIST 和 NSA 开发的，于 1993 年作为美国联邦信息处理标准（FIPS PUB 180）公布，1995 年修订为 FIPS PUB 181，1995 年修订为 SHA-1。另外还有 4 种变体曾经发布，以提升输出的范围和变更一些细微设计，分别是 SHA-224、SHA-256、SHA-384 和 SHA-512（名称中的数字代表摘要长度）。

SHA 基于 MD4 算法，SHA 的设计很近似于 MD4 模型。SHA 在用于数字签名的标准算法（DSS）中也是安全性很高的一个哈希算法。该算法的输入为小于 264b 长的任意消息，分为 512b 长的分组，输出为 160b 长的消息摘要。因为它能产生 160b 的哈希值，所以，抗穷举攻击能力更强。

SHA 与 MD5 都是来自 MD4，所以它们有许多相似之处。这里对 SHA 的细节不进行介绍，仅在表 5.2 中对这两种报文摘要算法进行比较。

总之，SHA-1 比 MD5 抗击穷举搜索的强度高，但执行速度较慢。

表 5.2　MD5 与 SHA-1 的比较

算　　法	摘要长度/b	最大报文长度	分组处理长度/b	运算次数	常数个数
MD5	128	无限制	512	64（4 轮 16 次）	64
SHA-1	160	$2^{64}-1$	512	80（4 轮 20 次）	4

5.2　消息认证码

如前所述，采用消息摘要好像是为了进行完整性认证。但是，由于哈希函数都是没有加密功能的，所以在消息传送时，用了对称密钥加密。这样，加密级别低就避免不了中间侵入者将原消息进行篡改，重新生成一个摘要后再发给接收者，使接收者无法进行真实性认证，完整性认证也难于保证。

5.2.1　MAC 函数

针对消息摘要的不足，一个基本的改良想法是发送者采用不对称密钥对摘要加密。这样，不仅使得完整性认证得以保证，也因为采用了含有身份信息的不对称密钥进行了加密，使得真实性认证也得以进行。这样，附加在原消息上用以及进行消息认证的哈希码就称为消息认证码，具备这样功能的散列函数称为 MAC 函数。

MAC 函数是消息 M 和不对称密钥 K 的函数，即 MAC = $C(M, K)$ 或 MAC = $C_K(M)$。

如果要画出带有 MAC 的消息传输与认证示意图，只需要将图 5.2 中的 H 改为 MAC 即可。这一改，不仅意味着生成了一个消息认证码 MAC，而且意味着这个 MAC 是用了发送方的私钥加密的。需要注意，从形式上看，MAC 函数与对称加密函数极为类似，都需要一个信源端和信宿端共享的密钥。但是，它们又有本质上的区别。

（1）加密算法要求可逆性，而 MAC 算法不要求可逆性。

（2）加密函数明文长度与密文长度一般相同，是一对一的函数，而 MAC 函数则是多对一的函数，其定义域由任意长的消息组成，而值域则由 MAC 比特的比特位构成。若报文长度为 m 位，MAC 长度为 n 位，则有 2^m 种报文对应 2^n 种 MAC。由于 $m >> n$，所以一定存在不同的报文产生相同的 MAC。例如，使用 100b 的报文和 10b 的 MAC，则有 2^{100} 种报文对应 2^{10} 种 MAC，平均而言，$2^{100}/2^{10} = 2^{90}$ 个报文具有相同的 MAC。

（3）MAC 函数比加密函数更不容易被攻破，因为即便被攻破，也无法认证其正确性。

5.2.2　安全 MAC 函数的性质

一个安全的 MAC 函数应具备下列性质。

（1）对于已知的 M 和 MAC，无法找到满足 MAC′ = MAC 的另一个 M'。

（2）$C_K(M)$ 应当是均匀分布的，则对于任何随机选择的报文 M 和 M'，找到 $C_K(M') = C_K(M)$ 的概率是 2^{-n}，n 为 MAC 的位数。

（3）若 M' 是 M 的一个已知变换，则找到 $C_K(M') = C_K(M)$ 的概率是 2^{-n}。

因此，若只有收发双方才知道密钥 K，并且接收到的 MAC 与计算出的 MAC 相等，则接收方可以相信：

（1）接收到的消息未被修改。

（2）接收到的消息来自真正的发送方。

（3）如果消息中含有序列号，则消息的顺序也是正确的。

5.2.3　CBC-MAC

CBC-MAC 也称为数据认证算法，是建立在 DES 基础上，基于分组密码，并按照密文块链接（Cipher Block Chaining，CBC）模式操作的 MAC 构造方法之一，也是 ANSI 的一个标准，如图 5.5 所示。CBC 模式的基本特点如下。

（1）将要认证的数据分成连续的 64b 的分组 D_1、D_2、…、D_N，若最后的分组不足 64b，在其后加 0，补足 64b。

（2）每个分组在用密钥加密之前要先与前一组的密文组进行异或运算生成其加密过程的种子向量。初始向量 $\boldsymbol{O}_0$ 取零。

（3）最后的 MAC 用 $\boldsymbol{O}_N$ 最左边 M 位表示，并且 $16 \leqslant M \leqslant 64$。

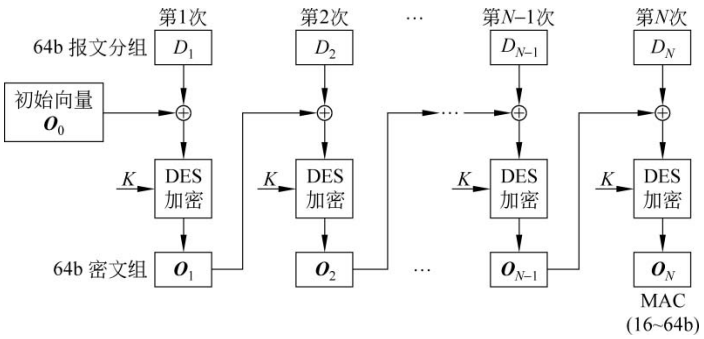


图 5.5　数据认证算法

5.3　数 字 签 名

5.3.1　数字签名及其特征

1. 数字签名的概念

在日常生活中，为了确认一件作品及其源出处，常需要采取签名、落款、骑缝章等手段，以便于鉴别。在数据通信过程中，有时会发生一方对另一方的消息的假冒、伪造、篡改和对于发送消息行为的抵赖等欺骗行为。

在前两节中已经解决了对是否有冒充和伪造的真实性认证以及对于篡改的完整性认证问题。但对于否认（抵赖）的认证问题还没有解决。于是人们从现实世界中的签名盖章想到了数字签名（digital signature）。

在 ISO 7498-2 标准中，数字签名定义为：“附加在数据单元上的一些数据，或是对数据单元所做的密码变换，这种数据和变换允许数据单元的接收者用于确认数据单元来源和数据单元的完整性，并保护数据，防止被人（例如接收者）进行伪造。”

显然，这个定义不仅单一地去解决抵赖性认证问题，把完整性和真实性也囊括了进来。

如表 5.3 所示，数字签名是完整性和真实性认证的补充。

表 5.3　报文摘要、消息认证码和数字签名对消息保护目标的差异

	完整性	真实性	不可抵赖性	对摘要加密	加密强度
报文摘要（MD）	是	否	否	否	无
消息认证码（MAC）	是	是	否	是	对称
数字签名	是	是	是	是	非对称

2. 数字签名与手工签名的特点

（1）签名是可信的。
（2）签名是无法被伪造的。
（3）签名不可以重复使用。
（4）签名以后不可以被篡改。
（5）签名具有不可否认性。

3. 对数字签名的要求

从有效性和可行性出发，对数字签名技术有以下要求。
（1）签名的结果必须是与签名的报文相关的二进制位串。
（2）签名要能够认证签名者的身份以及签名的日期和时间。
（3）签名能够用于证实被签报文的内容的真实性。
（4）签名的产生、识别和认证应比较容易。
（5）数字签名应当可以备份。
（6）用已知的签名构造一个新的报文或由已知的报文产生一个假冒的签名，在计算上都是不可行的。
（7）签名可以由第三方认证，以解决双方在通信中的争议。

5.3.2　直接数字签名

直接数字签名就是签名过程只有发送方和接收方参与。实施这种方法的前提是接收方可以通过某种方式认证发送方提交的凭证，也可以在发生争议时将该凭证交第三方仲裁。那么，用什么作为直接数字签名的凭证呢？

不对称密钥就可以作为数字签名中的凭证。例如，发送方 A 和接收方 B 使用只有双方才使用的不对称密钥。那么，A 私钥对消息进行签名发送；B 收到消息后，使用配对的公钥对签名进行认证。如果认证通过，说明消息就是 A 发送的，因为只有 A 采用配对的私钥。这个过程中，A 不可能冒充第三方，也无法否认自己的发送；B 也无法篡改和伪造。但是，非对称密钥算法的效率是很低的，不宜用于长报文的加密。为此可以采用鉴别码，将报文 *M* 通过一个单向哈希函数生成短的定长鉴别码，将认证与签名结合起来进行。

直接签名后的报文还有可能被接收方滥用。例如，A 发送给 B 一张电子支票，有可能被 B 多次复制兑换现金。如果 A 在报文中再增加一种特有凭证，如时间戳（timestamp），就可以避免这种情况发生。

此外，直接签名方法将一种算法既用于认证又用于签名，使签名的有效性完全依赖于

密钥体制的安全性，也会形成一些漏洞。例如，发送方会声称自己的密钥被窃或被盗用，来否认已经发送报文。为避免这样的威胁，可以将每一个被签名的报文中再包含一个时间戳，标明报文发送的日期和时间，同时要求一旦密钥丢失或被盗用，要立即向管理机构报告并更换密钥。但是，若密钥真正被盗，盗窃者可以伪造一个报文，并加上一个他盗窃密钥之前的时间戳，也还是不容易被发现。

5.3.3　有仲裁的数字签名

有仲裁的数字签名的基本思想是：发送方完成签字后，不是直接发送给接收方，而是将报文和签字先发送给双方共同信任的第三方进行认证，第三方认证无误后，再附加一个“已经通过认证”的说明并注上日期，一同发送给接收方。由于第三方的介入，发送方和接收方都无法抵赖。

有仲裁的数字签名方法也有很多，下面仅举几例。假定报文由 X 向 Y 传送，A 为仲裁者，M 为传输报文，$H(M)$为哈希函数值，||为链接，$\mathrm{ID_X}$ 为 X 的身份码，T 是时间戳，则可以有如下 3 种方案。

1. 方案 1

1）认证步骤

① X→A：$M \| E_{K_{\mathrm{XA}}}[\mathrm{ID_X} \| H(M)]$。X 将签名 $E_{K_{\mathrm{XA}}}[\mathrm{ID_X} \| H(M)]$和报文 M 发给 A。K_{XA} 为 X 和 A 的共享密钥。

② A→Y：$E_{K_{\mathrm{AY}}}[\mathrm{ID_X} \| M \| E_{K_{\mathrm{XA}}}[\mathrm{ID_X} \| H(M)] \| T]$。A 对签字认证后，再附加上时间戳 T，并用 A 和 Y 共享的密钥 K_{AY} 加密后转发给 Y。

③ Y 收到 A 发来的报文，解密后，将结果保存起来。由于 Y 不知道 K_{XA}，所以不能直接检查 X 的签字，只能相信 A。

当出现争议时，Y 可以声称自己收到的 M 来自 X，并将

$$E_{K_{\mathrm{AY}}}[\mathrm{ID_X} \| M \| E_{K_{\mathrm{XA}}}[\mathrm{ID_X} \| H(M)] \| T]$$

发送给 A，让 A 仲裁。

2）特点

这个方案是建立在 X 和 Y 都对 A 高度信任的基础上的，即：

（1）X 相信 A 不会泄露 K_{XA}，也不会伪造自己的签名。

（2）Y 相信 A 所认证 X 的签字是可靠的。

（3）X 和 Y 都相信出现争议时 A 能公正地处理。

3）结论

（1）X 相信 Y 无法对收到的报文予以否认。

（2）Y 相信 X 不会对他所发送的报文予以否认。

但是，这个方案未提供保密性，X 传送给 A 的 M 是明文形式。此外，方案本身没有对仲裁者的限制机制，一旦仲裁者不公正，如与发送方共谋否认发送过的报文，或与接收方连手伪造发送方的签字，都会形成签字的漏洞。

2. 方案 2

认证步骤：

① X→A：$ID_X \| E_{K_{XY}}[M] \| E_{K_{XA}}[ID_X \| H(E_{K_{XY}}[M])]$。

② A→Y：$E_{K_{AY}}[ID_X \| E_{K_{XY}}[M] \| E_{K_{XA}}[ID_X \| H(E_{K_{XY}}[M])] \| T]$。

这个方案用 X 和 Y 的共享密钥 K_{XY} 加密所传送的 M，从而提供了保密性。但还没有解决对仲裁者的约束。

3. 方案 3

认证步骤：

① X→A：$ID_X \| E_{SK_X}[ID_X \| E_{PK_Y}[E_{SK_X}[M]]$。

② A→Y：$E_{SK_A}[ID_X \| E_{PK_Y}[E_{SK_X}[M]] \| T]$。

在这个方案中，仲裁者只能用 X 的公钥对 $E_{SK_X}[ID_X \| E_{PK_Y}[E_{SK_X}[M]]$ 解密，得到 ID_X' 与以明文形式传送来的 ID_X 进行比较，确认这个报文确实来自 X，却不能解密 $E_{PK_Y}[E_{SK_X}[M]]$。$E_{PK_Y}[E_{SK_X}[M]]$要由 Y 才能解密。因为 Y 可以使用 PK_A 解密 $E_{SK_A}[ID_X \| E_{PK_Y}[E_{SK_X}[M]] \| T]$，进一步用 SK_Y 解密 $E_{PK_Y}[E_{SK_X}[M]] \| T]$，再用 PK_X 解密 $E_{SK_X}[M]$。这样就使仲裁方无法与任何一方共谋。

5.3.4 数字签名算法

数字签名算法（Digital Signature Algorithm，DSA）是美国国家标准委员会（NIST）公布的数字签名标准（Digital Signature Standard，DSS）。DSA 最早公布于 1991 年，在征求公众意见后进行了两次修改，又分别于 1993 年和 1996 年发表。图 5.6 描述了 DSA 签名的基本过程。

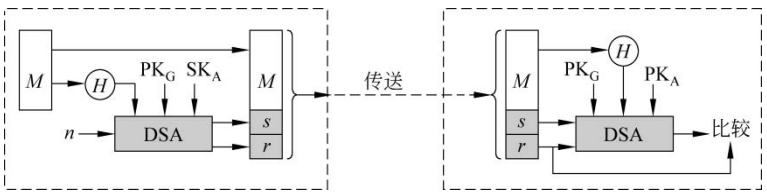


图 5.6　DSA 签名的基本过程

DSA 算法的签名函数以以下参数作为输入。

（1）用 SHA 方法生成的报文摘要。

（2）一个随机数。

（3）发送方的私有密钥 SK_A。

（4）全局公钥 PK_G——供所有用户使用的一族参数。

DSA 算法输出两个数据：s 和 r。这两个输出就构成了对报文 M 的数字签名。

接收方收到报文后，先产出报文的摘要，再将这个摘要和收到的签名以及全局公钥 PK_G、发送方的公开密钥 PK_A 一起送到 DSA 的认证函数中，生成一个新的 r'。若 r' 与 r 相等，就说明签字有效。

DSA 算法的安全性不再依赖于加密密钥的安全性。同时，其计算基于求离散对数的困难性，使攻击者从 r 恢复 n，或从 s 恢复 SK_A 都是在计算上不可行的。所以，DSA 比采用 RSA 的签名方法要可靠得多。

除了基于离散对数的签名算法外，人们还开发了其他一些签名算法。这里不再介绍。

5.4　基于凭证比对的身份认证

5.4.1　生物特征身份认证

生物特征身份凭证一般采用用户固有的生物特征和行为特征，要求这些具有唯一性和永久性。下面介绍几种主要的生物身份凭证及其认证方法。

1. 指纹

指纹是历史最为悠久的生物身份凭证。据著名指纹专家刘持平先生论证，早在 7000 年前我们的祖先就开始进行指纹识别的研究。到了春秋战国时代，手印检验不仅广泛应用于政府和民间的书信与邮件往来之中，并已经开始用于侦讯破案之中。

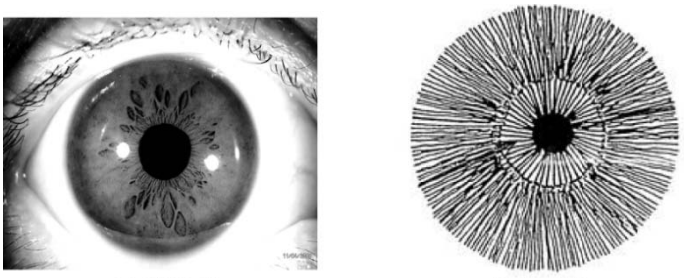
图 5.7　指纹的细节特征

指纹是一种十分精细的拓扑图形。如图 5.7 所示，一枚指纹不足方寸，上面密布着 100～120 个特征细节，这么多的特征参数组合的数量达到 640 亿种（英国学者高尔顿提出的数字）。并且由于它从胎儿 4 个月时生成后保持终生不变，因此，用它作为人的唯一标识，是非常可靠的。

指纹识别主要涉及 4 个过程：读取指纹图像、提取指纹特征、保存数据和比对。目前已经开发出计算机指纹识别系统，可以比较精确地进行指纹的自动识别。

2. 虹膜

虹膜是位于眼睛黑色瞳孔与白色巩膜之间的环形部分（图 5.8（a））。从外观上看，它由许多相互交错的类似斑点、细丝、冠状、条纹、隐窝、皱褶、色素斑等构成（图 5.8（b）），是人体中最独特的结构之一。这些细微特征信息也被称为虹膜的纹理信息，主要由胚胎发



(a) 虹膜位置　　(b) 虹膜结构

图 5.8　眼睛与虹膜