第 5 章　数字签名技术与应用

本章导读：

数字签名是信息安全的一个非常重要的分支，它在大型网络安全通信中的密钥分配、安全认证、公文安全传输以及电子商务系统中的防否认等方面具有重要作用。

数据在不安全的网络上进行传输时，为确保传输数据的安全性，必须采取一系列的安全技术，如加密技术、数字签名技术、身份认证技术、信息隐藏技术等，其中数字签名技术可以保证数据交换的完整性、发送信息的不可否认性，并可对交易者身份进行有效认证，是当前电子商务、电子政务安全的必备技术之一。

数字签名（也称电子签名）的思想与手写签名相似，要保证签发的消息不可伪造、保证收发双方的不可否认，并可经过第三方验证数据传输过程。但电子签名与手写签名也有很大区别，电子签名采用密码学理论与技术保证这一过程，在网络空间具有更高的安全性。

本章主要介绍数字签名的基础原理、常用的签名算法以及国产的 SM9 算法，并介绍了数字签名标准及应用产品。

5.1　数字签名的基本原理

政治、军事、外交等领域的文件、命令和条约，商业中的契约，以及个人之间的书信等，传统上都采用手书签名或印章，以便在法律上能认证、核准和生效。随着计算机通信的发展，人们希望通过电子设备实现快速、远距离的交易，数字（或电子）签名便应运而生，并开始用于商业通信系统，如电子邮递、电子转账和办公自动化等系统中。

类似于手书签名，数字签名也应满足以下要求。

（1）收方能够确认或证实发方的签名，但不能伪造。

（2）发方发出签名的消息送收方后，就不能再否认他所签发的消息。

（3）收方对已收到的签名消息不能否认，即收到认证。

（4）第三者可以确认收发双方之间的消息传送，但不能伪造这一过程。

5.1.1　数字签名与手书签名的区别

数字签名与手书签名的区别在于，手书签名是模拟的且因人而异。数字签名是 0 和 1 的数字串，因消息而异。数字签名与消息认证的区别在于，消息认证使收方能验证消息发送者及所发消息内容是否被篡改过。当收发者之间没有利害冲突时，这对于防止第三者的破坏来说是足够了。但当收者和发者之间有利害冲突时，单纯用消息认证技术就无法解决他们之间的纠纷，此时需借助数字签名技术。

为了实现签名目的，发方须向收方提供足够的非保密信息，以便使其能验证消息的签名。但又不泄露用于产生签名的机密信息，以防止他人伪造签名。因此，签名者和证实者可公用的信息不能太多。任何一种产生签名的算法或函数都应当提供这两种信息，而且从公

开的信息很难推测出用于产生签名的机密信息。另外,任何一种数字签名的实现都有赖于精心设计的通信协议。

5.1.2 数字签名的分类

数字签名有两种,一种是对整个消息的签名,一种是对压缩消息的签名,它们都是附加在被签名消息之后或某一特定位置上的一段签名图样。若按明、密文的对应关系划分,每一种又可分为两个子类。一类是确定性数字签名,其明文与密文一一对应,它对特定消息的签名不变化(使用签名者的密钥签名),如 RSA、ElGamal 等签名;另一类是随机化的或概率式数学签名,它对同一消息的签名是随机变化的,取决于签名算法中的随机参数和取值。

一个签名体制一般含有两个组成部分,即签名算法和验证算法。对 M 的签名可简记为 $\mathrm{Sig}(M)=s$(有时为了说明密钥 k 在签名中的作用,也可以将签名写成 $\mathrm{Sig}_k(M)$ 或 $\mathrm{Sig}(M,k)$,而对 s 的证实简记为 $\mathrm{Ver}(s)=\{真,伪\}=\{0,1\}$。签名算法或签名密钥是秘密的,只有签名人掌握。证实算法应当公开,以便于他们进行验证。

一个签名体制可由量 (M,S,K,V) 表示,其中 M 是明文空间,S 是签名的集合,K 是密钥空间,V 是证实函数的值域,由真、伪组成。

对于每一 $k\in K$,有一签名算法,易于计算 $s=\mathrm{Sig}_k(m)\in S$。利用公开的证实算法,即

$$\mathrm{Ver}_k(s,m)\in\{真,伪\}$$

可以验证签名的真伪。

它们对每一 $m\in M$,真签名 $\mathrm{Sig}_k(m)\in S$ 为 $M\to S$ 的映射。易于证实 S 是否为 M 的签名。

$$\mathrm{Ver}_k(s,m)=\begin{cases}真, & 当\ \mathrm{Sig}_k(s,m)\ 满足验证方程\\ 伪, & 当\ \mathrm{Sig}_k(s,m)\ 不满足验证方程\end{cases}$$

体制的安全性在于,从 m 和其签名 s 难以推出 k,或伪造一个 m',使 $\mathrm{Sig}_k(m')$ 满足验证方程。

消息签名与消息加密有所不同,消息加密和解密可能是一次性的,它要求在解密之前是安全的,而一个签名的消息可能作为一个法律上的文件(如合同等)很可能在对消息签署多年之后才验证其签名,且可能需要多次验证此签名。因此,签名的安全性和防伪造的要求会更高,且要求证实速度比签名速度要快些,特别是联机在线时进行实时验证。

5.1.3 使用数字签名

随着计算机网络的发展,过去依赖于手写签名的各种业务都可用这种电子化的数学签名代替,它是实现电子贸易、电子支票、电子货币、电子出版及知识产权保护等系统安全的重要保证。数字签名已经并将继续对人们如何共享和处理网络上信息以及事务处理产生巨大的影响。

例如,在大多数合法系统中对大多数合法的文档来说,文档所有者必须给一个文档附上一个时间标签,指明文档签名对文档进行处理和文档有效的时间与日期。在用数字签名对文档进行标识之前,用户可以很容易地利用电子形式为文档附上电子时间标签。因为数字签名可以保证这一日期和时间标签的准确性和证实文档的真实性,数字签名还提供了一个额外的功能,即它提供了一种接收者可以证明确实是发送者发送了这一消息的方法。

使用电子汇款系统的人也可以利用电子签名。例如,假设有一个人要发送从一个账户到另一个账户转存 10 000 美元的消息,如果这一消息通过一个未加保护的网络,那么“黑

客”就能改变资金的数量从而改变了这一消息。但是，如果发送者对这一消息进行数字签名，由于接收系统核实错误，从而识别出对此消息的任何改动。

大范围的商业应用要求变更手写签名方式时，可以使用数字签名。其中一例便是电子数据交换(Electronic EDI)。EDI 是商业文档消息的交换机制。美国联邦政府用 EDI 技术来为消费者购物提供服务。在 EDI 文档里，数字签名取代了手写签名，利用 EDI 和数字签名，只需通过网络介质(Data Interchang)，即可进行买卖并完成合同的签订。

数字签名的使用已延伸到保护数据库的应用中。一个数据库管理者可以配置一套系统，它要求输入消息到数据库的任何人在数据库接收之前必须数字化标识该消息。为了保证真实性，系统也要求用户标识对消息所做的任何修改。在一个用户查看已被标识过的消息之前，系统将核实创建者或编辑者在数据库消息中的签名，如果签名核实结果正确，用户就知道没有未经授权的第三者改变这些消息。

5.2 RSA 签名

安全参数：令 $n=pq$，p 和 q 是大素数，选 e 并计算出 d，使 $ed=1 \bmod (p-1)(q-1)$，公开 n 和 e，将 p、q 和 d 保密。则所有的 RSA 参数为 $k=(n,p,q,e,d)$。

数字签名：对消息 $M\in Z_n$ 定义

$$S = \mathrm{Sig}(M) = M^d \bmod n$$

为对 M 的签名。

签名验证：对给定的 M、S 可按下式验证：设 $M'=S^e \bmod n$，如果 $M=M'$，则签名为真，否则，不接受签名。

显然，由于只有签名者知道 d，由 RSA 体制可知，其他人不能伪造签名，但容易证实所给任意(M,S)对是不是消息 M 和相应的签名所构成的合法对。RSA 体制的安全性依赖于分解的困难性。

ISO/IEC 9796 和 ANSI X9.30—199X 已建议将 RSA 作为数字签名的标准算法。PKCS#1 是一种采用杂凑算法(如 MD2 或 MD5 等)和 RSA 相结合的公钥密码标准。

5.3 ElGamal 签名

该体制由 T. ElGamal 在 1985 年给出，其修正形式已被美国 NIST 作为数字签名标准(DSS)，它是 Rabin 体制的一种变形。此体制专门为签名而设计，方案的安全性基于求离散对数的困难性。可以看出，它是一种非确定性的双钥体制，即对同一明文消息，由于随机参数选择的不同而有不同的签名。

1. 体制参数

p：一个大素数，可使 Z_p 中求解离散对数为困难的问题。

g：是 Z_p 中乘群 Z_p^* 的一个生成元或本原元素。

M：消息空间为 Z_p^*。

S：签名空间为 $Z_p^* \times Z_{p-1}$。

X：用户密钥 $x\in Z_p^*$，公钥为 $y=g^x \bmod p$。

安全参数为：$k=(p,g,x,y)$，其中 p、g、y 为公钥，x 为秘密钥。

2. 签名过程

给定消息 M，发送端用户进行下述工作。

(1) 选择秘密随机数 $k\in Z_p^*$。

(2) 计算压缩值 $H(M)$，并计算

$$r = g^k \bmod p$$

$$s = (H(M) - xr)k^{-1} \bmod (p-1)$$

(3) 将 $\mathrm{Sig}(M,k)=(M,r,s)$ 作为签名，将 (M,r,s) 送给对方。

3. 验证过程

收信人收到 (M,r,s)，先计算 $H(M)$，并按下式验证签名，即

$$y^r r^s = g^{H(M)} \bmod p$$

这是因为 $y^r r^s = g^{rx} g^{sk} = g^{(rx+sk)} \bmod p$，由上式有 $(rx+sk)=H(M) \bmod (p-1)$。

在此方案中，对同一消息 M，由于随机数 k 不同而有不同的签名值 (M,r,s)。

4. 安全性

它依赖于解离散对数问题的困难性。ANSI X9.30—199X 已将 ElGamal 签名体制作为签名标准算法。

5.4 SM9 算法

SM9 算法是基于对的标识密码算法，包含 4 个部分，即数字签名算法、密钥交换协议、密钥封装机制和公钥加密算法。在这些算法中使用了椭圆曲线上的对这一个工具，不同于传统意义上的 SM2 算法，可以实现基于身份的密码体制，也就是公钥与用户的身份信息即标识相关，从而比传统意义上的公钥密码体制有更多优点，省去了证书管理等。其中，数字签名算法适用于接收者通过签名者的标识验证数据的完整性和数据发送者的身份，也适用于第三方确定签名及所签数据的真实性。密钥交换协议可以使用通信双方通过双方的标识和自身的私钥经过两次或者可选 3 次信息传递过程，计算获取一个由双方共同决定的共享秘密密钥。密钥封装机制和公钥加密算法中，利用密钥封装机制可以封装密钥给特定的实体。公钥加密和解密算法即基于标识的非对称秘密算法，该算法使消息发送者可以利用接收者的标识对消息进行加密，唯有接收者可以用相应的私钥对该密文进行解密，从而获取消息。

SM2 中的总则部分同样适用于 SM9，由于 SM9 总则中添加了适用于对的相关理论和实现基础。椭圆曲线双线性对定义和计算在扩域上进行，总则中给出了扩域的表示和运算，数据类型转换同样包括整数与字节串、比特串和字节串、字节串和域元素、点和字节串之间的转换，其中字节串和域元素之间的数据类型转换涉及扩域。系统参数的生成比 SM2 复杂，涉及对的相关参数，验证也复杂。

5.4.1 SM9 加密算法

加密算法由以下 4 个算法构成，即建立(Setup)、密钥提取(KeyGen)、加密(Encrypt)和解密(Decrypt)。

Setup(k)→PK，MSK：建立算法以安全参数 k 为输入，输出为公共参数 PK 和主密

钥 MSK。

KeyGen(PK,MSK,ID)→SK_{ID}：密钥提取算法以公共参数 PK、主密钥 MSK 和一个身份信息 ID 为输入，输出该身份信息对应的私钥 SK_{ID}。并通过安全信道将 SK_{ID}返回给对应用户。

Encrypt(PK,M,SK_{ID})→CT：加密算法以公共参数 PK、明文消息 M 以及接收者的身份信息 ID 为输入，输出密文 CT。该算法由信息发送者(加密者)完成，并将密文通过公开信道发送给对应的接收者(解密者)。

Decrypt(PK,CT,SK_{ID})→M：解密算法以公共参数 PK、密文 CT、私钥 SK_{ID}为输入，在正确解密时输出明文 M，或在不能正确解密时返回符号⊥。该算法由信息接收者(解密者)完成。

算法必须满足正确性约束条件，即对于给定的身份信息 ID 和与之对应的私钥 SK_{ID}，有 $\forall M$:Decrypt(PK,Encrypt(PK,ID,M),SK_{ID})=M。

5.4.2 SM9 身份认证

身份认证是网络安全的重要机制之一，也是实现身份信息保密的重要技术。传统身份认证应用系统采用用户名加口令方式实现身份验证，网络之间的信息传输都是明文。这种传统的认证方式存在很多的安全隐患，信息极易泄露。用户为了便于记忆，其用户名和密码往往过于简单且带有一定的规律性，易被猜测、易泄露；同时用户在输入密码时易被偷窥，而密码在传输过程中也易被黑客截获；信息以明文形式传输，或密文的加密强度太低，很容易破解；而使用 PKI/CA 证书体系的身份认证机制，需要事先申请证书，这对用户来说申请过程繁琐、使用复杂，对应用商来说开发难度大、部署困难，难以推广。

而基于身份标识的 SM9 算法通过用户的手机号码或邮件地址作为标识，简单易用，认证过程中没有任何用户名和密码传递，安全可靠。其工作原理主要采用了挑战/应答模式的 CHAP 认证协议，简称 CHAP(CHallenge Authentication Protocol)。该协议基于签名的挑战/应答，可以抵抗木马、口令字典等攻击，如图 5-1 所示。

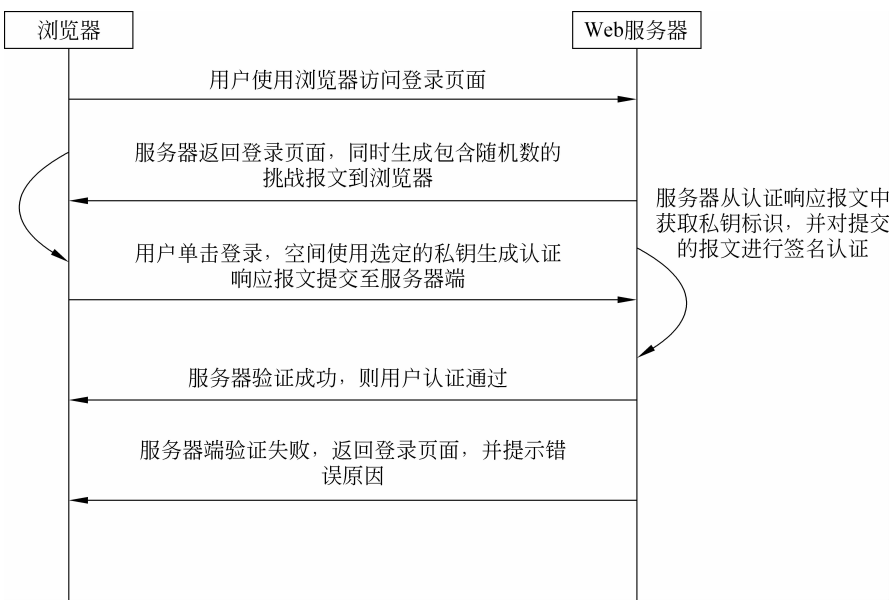

图 5-1 SM9 算法身份认证过程

传统的登录界面是使用用户名和密码，而使用 SM9 提供的登录方式只需输入 KEY 的 PIN 码即可登录。PIN 码可以设置错误次数，登录错误超过设定次数即可自锁。

5.4.3 传统的 PKI 体系与 IBC 体系的对比

现在国际上普遍使用 IBC(Identity-Based Cryptography，标识密码算法)，IBC 是新兴的密码技术。在标识的密码系统中，每个实体具有一个有意义的、唯一的标识，如姓名、IP 地址、电子邮箱地址、手机号码等，这个标识本身就是实体的公钥。无须预先协商密码或者交换证书，可以大大减少传统证书体系中申请和验证环节，易于使用。用户的私钥由密钥生成中心(Key Generate Center，KGC)根据系统主密钥和用户标识计算得出，基于身份的标识密码是传统的 PKI 证书体系的最新发展(图 5-2)，而 SM9 算法就是国家密码局对国家标识密码体系 IBC 标准的规范，并于 2007 年 12 月 16 日给予国家 IBC 标准 SM9 商密算法型号(图 5-3)。SM9 算法是国家商用密码管理局颁布的合规性算法，可达到相当于 RSA 3072 位加密强度，破解需要大约 2500 亿台高性能计算机计算 10 亿年。

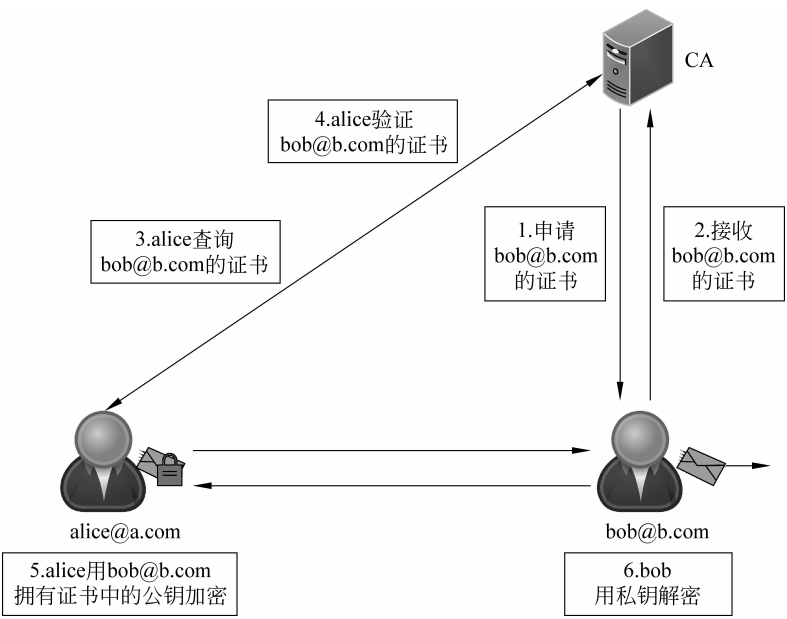

图 5-2 PKI 加密过程

由图 5-2 和图 5-3 可以看出，PKI 体系与 IBC 体系的应用区别，具体见表 5-1。

表 5-1 PKI 与 IBC 体系的区别

PKI 体系	IBC 体系
公钥是随机数	公钥可以是邮箱地址
通过证书将用户的公钥与身份关联起来	公钥即用户的身份标识
信息发送方必须获得接收方的公钥证书	信息发送方只需要获知接收方的身份标识(如姓名、IP 地址、电子邮箱地址、手机号码等)
证书颁发和管理系统复杂难以部署	无需颁发和管理证书

续表

PKI 体系	IBC 体系
每次发送信息之前，都需要与管理中心通信交互，验证证书的有效性	可以本地离线加、解密
难以实现基于属性、策略的加密	可增加时间或固定 IP 等方式解密信息的安全策略控制
存放的收信方证书随发送邮件数量的增大而增多，在线通信交互越繁忙，管理负担和管理成本会同比例放大	发送邮件数量级增大，管理负担和管理成本的增加并不明显
实现成本高，效率低下	实现成本低，效率较高
系统运行维护成本高	运营管理方便，成本低

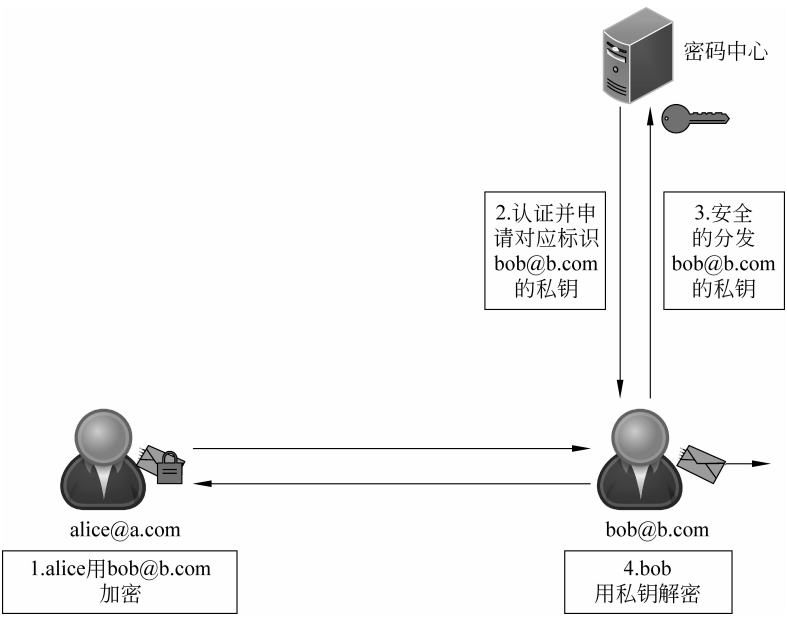

图 5-3　IBC 加密过程

由图 5-2 和图 5-3 的分析可见，与 PKI 体系相比，IBC 体系的应用优势主要表现在以下几个方面。

- 无需 PKI 体系中的数字证书，无需证书颁发机构 CA 中心，无需证书的发布与查询，使用简单，部署方便，尤其适用于海量用户的安全系统。
- 无需 PKI 中证书验证等计算过程，具备较低的计算代价，适用于手机终端。
- 无需 PKI 的在线连接 CA 服务器查询与验证证书状态，具备较低的通信代价。
- 丰富的策略控制机制，将身份认证与访问控制合二为一。

5.4.4　SM9 算法的应用

国际上现在普遍使用 IBC 来解决邮件安全问题。例如，美国 HP 公司的 IBC 加密电子邮件的企业用户已近千家，涉及银行、零售、保险、能源、医疗保健和政府等行业，使用加密邮

件用户数量已近数千万，其云端系统年处理超过 12 亿封邮件的加、解密；近 30%财富 2000 企业都使用其 IBC 技术来实现邮件加密；美国微软、中国台湾趋势科技、SendMail、Proofpoint 等均提供基于 IBC 的安全邮件产品。使用 IBC 来实现的邮件安全方案，用户直接使用邮件地址作为公钥，无须预先协商密码或者交换证书，安全性和易用性得到很好的结合。

在国内，国家密码管理局已经对 IBC 标识密码算法进行标准化，颁发了商密算法型号，即 SM9(商密第九号算法)。SM9 算法无须申请数字证书，在电子邮件加密方面具有不可替代的优势，其加密算法强度足以让现有计算能力无法破解。所以国内相关单位纷纷基于此算法推出邮件安全产品，如奇虎 360 的加密邮箱产品、深圳奥联的 Email 产品都使用了 SM9 算法，以及国内邮件厂商如 Coremail、安宁、亿邮、Fangmail 等均实现了 SM9 的支持，国家信息中心、中国信息安全测评中心也承担了 SM9 的应用示范试点项目，国家密码算法的应用将逐渐普及。

5.5 盲签名及其应用

为了说明盲签名的基本概念，本节假设 Alice 为消息拥有者，Bob 为签名人。在盲签名协议中，Alice 的目的是让 Bob 对某文件进行签名，但又不想让 Bob 知道文件的具体内容，而 Bob 并不关心文件中说些什么，他只是保证他在某一时刻以公正人的资格证实了这个文件。

Alice 从 Bob 处获得盲签名的过程一般有以下几个步骤。

(1) Alice 将文件 m 乘一个随机数得 m'，这个随机数通常称为盲因子，Alice 将盲消息 m'发送给 Bob。

(2) Bob 在 m'上签名后，将其签名 Sig(m')送 Alice。

(3) Alice 通过除去盲因子可从 Bob 关于 m'的签名 Sig(m')中得到 Bob 关于原始文件 m 的签名 Sig(m)。

D. Chaum 关于盲签名曾经给出一个非常直观的说明：盲签名就是先将要隐蔽的文件放进信封里，而除去盲因子的过程就是打开这个信封。当文件在一个信封中时，任何人都不能读它。对文件签名就是通过在信封里放一张复写纸，当签名者在信封上签名时，他的签名便透过复写纸签到了文件上。

下面所介绍的盲签名方案都是在 ElGamal 签名方案上构造的，其中 x 和 $y=a^x \bmod p$ 为签名者 Bob 的私钥和公钥。

5.5.1 盲消息签名

在盲消息签名方案中，签名者仅对盲消息 m'签名，并不知道真实消息 m 的具体内容。这类签名的特征是：Sig(m)=Sig(m')或 Sig(m)含 Sig(m')中的部分数据。因此，只要签名者保留关于盲消息 m'的签名，便可确认自己关于 m 的签名。

Alice　　Bob

选择消息 $m\in Z_p$，随机数 $h\in Z_{p-1}$。

计算 $\beta=a^h \bmod p$，$m'=mh \bmod (p-1)$ $\xrightarrow{(\beta,m')}$ 选择随机数 $k\in Z_{p-1}$

计算 $r=\beta^k \bmod p$，

$$S=(xr+m'k) \bmod (p-1)$$

$$\mathrm{Sig}(m)=(r,s) \xleftarrow{(r,s)}$$

验证方程为

$$a^s = y^r r^m \bmod p$$

从签名方程 $s=xr+m(k \bmod (p-1))$可知

$$a^s = y^r (a^k)^m = y^r a^{kmh} = y^r \beta^{km} = y^r r^m = y^r r^m \bmod p$$

因此，验证方程成立。

可以看出，在上述盲消息签名方案中，Alice 将 Bob 关于 m'的签名数据作为其对 m 的签名，即 $\mathrm{Sig}(m)=\mathrm{Sig}(m')$。所以，只要 Bob 保留 $\mathrm{Sig}(m')$，便可将 $\mathrm{Sig}(m)$与 $\mathrm{Sig}(m')$相联系。为了保证真实消息 m 对签名者保密，盲因子尽量不要重复使用。因为盲因子 h 是随机选取，所以，对一般的消息 m 而言，不存在盲因子 h，使 $m'(m'=mh \bmod (p-1))$有意义；否则，Alice 将一次从 Bob 处获得两个有效签名 $\mathrm{Sig}(m)$和 $\mathrm{Sig}(m')$，从而使得两个不同的消息对应相同的签名。这一点也是签名人 Bob 最不愿看到的。

盲消息签名方案在电子商务中一般不用于构造电子货币支付系统，因为它不保障货币持有者的匿名性。

5.5.2 盲参数签名

在盲参数签名方案中，签名者知道所签消息 m 的具体内容。按照签名协议的设计，签名收方可改变原签名数据，即改变 $\mathrm{Sig}(m)$而得到新的签名，但又不影响对新签名的验证。因此，签名者虽然签了名，却不知道用于改变签名数据的具体安全参数。

Alice　　Bob

选择 $m\in Z_p$，随机数 $h\in Z_{p-1}$。

计算 $\beta=a^h \bmod p \xrightarrow{(m,\beta)}$选择随机数 $k\in Z_{p-1}^*$

计算 $r'=\beta^k \bmod p$

$$s'=k^{-1}(m+xr')\bmod (p-1)$$

新签名 $\mathrm{Sig}(m)=(r,s)$，其中：$\xleftarrow{(r',s')} r=r', s=s'h^{-1} \bmod (p-1)$

验证方程为

$$r^s = a^m y^r \bmod p$$

在上述盲参数签名方案中，m 对签名者并不保密。当 Alice 对 $\mathrm{Sig}(m)$做了变化之后，(m,r,s)和(m,r',s')的验证方程仍然相同。

盲参数签名方案的这些性质可用于电子商务系统 CA 中心，为交易双方颁发口令。任何人虽然可验证口令的正确性，但包括 CA 在内谁也不知变化后的口令。在实际应用中，用户的身份码 ID 相当于 m，它对口令产生部门并不保密。用户从管理部门为自己产生的非秘密口令得到秘密口令的方法，就是将(ID,r',s')，转化为(ID,r,s)。这种秘密口令并不影响计算机系统对用户身份进行认证。另外，利用盲参数签名方案还可以构造代理签名机制中的授权人和代理签名人之间的授权方程，以用于多层 CA 机制中证书的签发以及电子支票和电子货币的签发。

5.5.3 弱盲签名

在弱盲签名方案中，签名者仅知 Sig(m')而不知 Sig(m)。如果签名者保留 Sig(m')及其他有关数据，待 Sig(m)公开后，签名者可以找出 Sig(m')和 Sig(m)的内在联系，从而达到对消息 m 拥有者的追踪。

Alice　　Bob

选随机数 a 和 b $\xleftarrow{r'}$选随机数 $k\in(1,p-1)$

计算 $r'=a^k \bmod p$

计算 $r=r'^a a^b \bmod p$

$m'=amr' r^{-1} \bmod q \xrightarrow{m'}$计算 $S'=r'x+km' \bmod q$

$s=(S'rr'^{-1}+mb) \bmod q \xleftarrow{s'}$

Sig(m)=(r,s)

验证方程为

$$a^s = y^r r^m \bmod p$$

在上述盲签名方案中，如果签名者 Bob 保留($m'r'S'$,k)，则当 Alice 公开 Sig(m)=(r,s)后，Bob 可求得 $a'=m'm^{-1} r'^{-1}r \bmod q$ 和 $b'=m^{-1}(S-S'rr'^{-1}) \bmod q$。

为了证实 Sig(m)=(r,s)是从 Sig(m')=($m'r'S'$)所得，Bob 只需验证等式 $r=r^{a'}a^{b'} \bmod p$ 是否成立，若成立，则可确认 $a'=a$、$b'=b$，从而确认 Sig(m)和 Sig(m')相对应。这充分说明上述方案的确是一个弱盲签名方案。

盲消息签名方案与弱盲签名方案的不同之处在于，后者不仅将消息 m 做了盲化，而且对签名 Sig(m')做了变化，但两种方案都未能摆脱签名者将 Sig(m)和 Sig(m')相联系的特性，只是后者隐蔽性更大一些。由此可以看出，弱盲签名方案与盲消息签名方案的实际应用较为类似。

5.5.4 强盲签名

在强盲签名方案中，签名者仅知 Sig(m')，而不知 Sig(m)。即使签名者保留 Sig(m')及其他有关数据，仍难以找出 Sig(m)和 Sig(m')之间的内在联系，不可能对消息 m 的拥有者进行追踪。

Alice　　Bob

(公钥(e,n))　　(密钥 d)

选择盲因子 r

计算 $m'=mr^e \bmod n \xrightarrow{m'}$计算盲签名

$\xleftarrow{s'}S'=(m')^d \bmod n$

计算签名，即

$$s = s'^{-1} \bmod n = m^d \bmod n$$

强盲签名方案是目前性能最好的一个盲签名方案，电子商务中使用的许多数字货币系统和电子投票系统的设计都采用了这种技术。