第6章　数字签名和认证协议

6.1　数字签名的基本概念

数字签名

- 数字签名以电子方式存储签名消息，是在数字文档上进行身份验证的技术。接收者和第三方能够验证文档来自签名者，并且文档签名后没有被修改，签名者也不能否认对文档的签名。
- 数字签名必须保障：
  - （1）接收者能够核实发送者对文档的签名；
  - （2）发送者事后不能否认对文档的签名；
  - （3）不能伪造对文档的签名

数字签名vs手写签名

- 一个签名有消息和载体两个部分，即签名所表示的意义和签名的物理表现形式。手写签名与数字签名之间存在着很大的差别，本质差别在于消息与载体的分割。

数字签名vs手写签名

- 传统签名中签名与文件是一个物理整体，
  - 具有共同的物理载体，
  - 物理上的不可分割、不可复制的特性
  - 签名与文件的不可分割和不能重复使用
- 数字签名中，签名与文件是电子形式
  - 没有固定的物理载体，即签名及文件的物理形式和消息已经分开
  - 电子载体是可以任意分割、复制的
  - 数字签名有可能与文件分割，被重复使用

数字签名vs手写签名

- 传统签名的验证是通过与存档手迹对照来确定真伪的，它是主观的、模糊的、容易伪造的，从而也是不安全的。
- 数字签名则是用密码，通过公开算法可以检验的，是客观的、精确的，在计算上是安全的。

数字签名vs手写签名

- 传统签名的基本特点：
  - 能与被签的文件在物理上不可分割
  - 签名者不能否认自己的签名
  - 签名不能被伪造
  - 容易被验证

数字签名vs手写签名

- 数字签名是传统签名的数字化，基本要求：
  - 能与所签文件“绑定”
  - 签名者不能否认自己的签名
  - 签名不能被伪造
  - 容易被自动验证

数字签名(digital signature)： 用于对数字消息签名,以防消息的伪造或篡改,也可用于通信双方的身份鉴别。

数字签名应具有的特性：

（1）签名是可信的：任何人可验证签名的有效性。

（2）签名是不可伪造的：除合法签名者外，其他人伪造签名是困难的。

（3）签名是不可复制的：一消息的签名不能复制为另一消息的签名。

（4）签名的消息是不可改变的：经签名的消息不能被篡改。

（5）签名是不可抵赖的：签名者事后不能否认自己的签名。

数字签名：对身份认证，保持数据完整性、不可否认性。

数字签名vs MAC

数字签名：对身份认证，保持数据完整性、不可否认性。

MAC可对身份认证，保持数据完整性，但不具有不可否认性。

RSA签名体制

体制参数：

选两个保密的大素数p和q，计算$n=p\times q$，$\phi(n)=(p-1)(q-1)$；

选一整数e，满足$1 < e < \phi(n)$，且$\gcd(\phi(n),e)=1$；

计算d，满足$d\cdot e\equiv 1 \bmod \phi(n)$；

以{e,n}为公开钥，{d,p,q}为秘密钥。

签名：

设消息为$m \in Z_n$，对其签名为

$$s \equiv m^d \bmod n$$

验证：

接收方在收到消息m和签名s后，验证

$$m \overset{?}{\equiv} s^e \bmod n$$

是否成立，若成立，则发送方的签名有效。

安全性：大数分解的困难性。

$$m \stackrel{?}{\equiv} s^e \bmod n$$

缺点：

（1）对任意y $\in Z_n$，任何人可计算$x \equiv y^e \bmod n$，因此任何人可伪造对随机消息x的签名。

（2）如果消息x_1和x_2的签名分别为y_1和y_2，则知道x_1, y_1, x_2, y_2的人可伪造对消息$x_1 x_2$的签名$y_1 y_2$。

（3）在RSA签名方案中，需签名的消息x $\in Z_n$，所以每次只能对$\lfloor \log_2 n \rfloor$位长的消息进行签名。签名速度慢。

克服缺陷的方法：　　签名之前先求消息的**Hash值**。

数字签名的一般描述：

明文消息：m　　私钥：x　　公钥：y

签名算法：$s=Sig_x(m)$

验证算法：$Ver_y(s, m)$

$$Ver_y(s,m)=\begin{cases} True & s=Sig_x(m) \\ False & s\neq Sig_x(m) \end{cases}$$

算法的安全性：从m和s难求密钥x，或伪造消息m'，使 m'和s可被验证为真。

6.2 数字签名标准

数字签名标准*DSS(Digital Signature Standard)*是由美国*NIST*公布的联邦信息处理标准*FIPS PUB* 186，最初于1991年公布，在考虑了公众对其安全性的反馈意见后，于1993年公布了其修改版。

ElGamal签名体制---- 基于离散对数问题

(1) 体制参数

p： 大素数；**q | p-1** 为大素数；

g： $g \in_R Z^*_p$，且$g^q \equiv 1 (\mathrm{mod}\ p)$，其中 $g \in_R Z^*_p$表示g是从Z^*_p中随机选取的，其中 $Z^*_p = Z_p - \{0\}$；

x： 用户A的私钥，$1<x<q$；

y： 用户A的公钥，$y \equiv g^x (\mathrm{mod}\ p)$。

原始的ElGamal签名体制中，$q = p-1$

(2) 签名：对于待签名的消息m，A执行以下步骤：

① 计算m的杂凑值H(m)。

② 选择随机数k：1<k<q，计算$r \equiv g^k (\bmod p)$。

$$s \equiv [k^{-1}(H(M)+xr)] mod\, q$$

③ 消息m的签名为 （r，s）

(3) 签名验证：接收方在收到消息m和签名(r, s)后，可以按照以下验证方程检验：

Ver(y, (r,s), m) = True 当且仅当

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

6.2.1 DSS的基本方式

首先将DSS与RSA的签名方式做一比较。RSA算法既能用于加密和签名，又能用于密钥交换。与此不同，DSS使用的算法只能提供数字签名功能。图6-2用于比较RSA签名和DSS签名的不同方式：

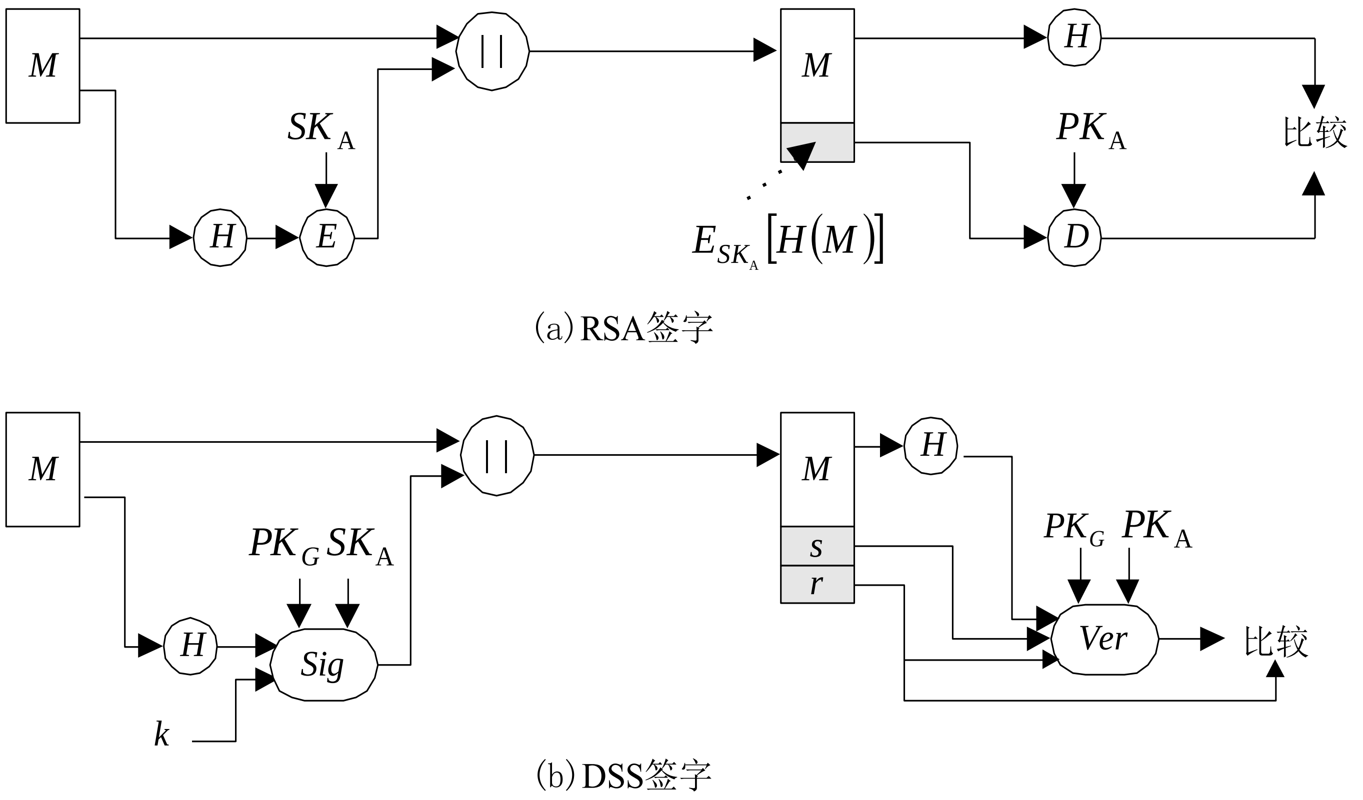


图6-2 RSA签名与DSS签名的不同方式

6.2.2 数字签名算法DSA

DSA安全性基于求离散对数的困难性。

算法描述如下：

6.2.2 数字签名算法DSA：DSA安全性基于求离散对数的困难性。

(1) 全局公开钥

p：$2^{L-1}<p<2^{L}$ 的大素数，
512≤L≤1024且L是64的倍数。

q：p-1的素因子，满足$2^{159}<q<2^{160}$ 。

g：$g \equiv h^{(p-1)/q} \bmod p$，h满足1<h<p-1
且使得$h^{(p-1)/q} \bmod p>1$的任一整数。

(2） 用户秘密钥x：

0<x<q的随机数。

(3) 用户的公开钥y：

$y \equiv g^{x} \bmod p$。

签名：

用户为待签消息m选取

秘密随机数$k, 0<k<q$

定义$Sig(m,k)=(r,s)$

$$r = g^{k} \bmod p \bmod q$$

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

$H(m)$:由SHA求出.

验证：

$$u_1 = H(m)s^{-1} \bmod q$$

$$u_2 = rs^{-1} \bmod q$$

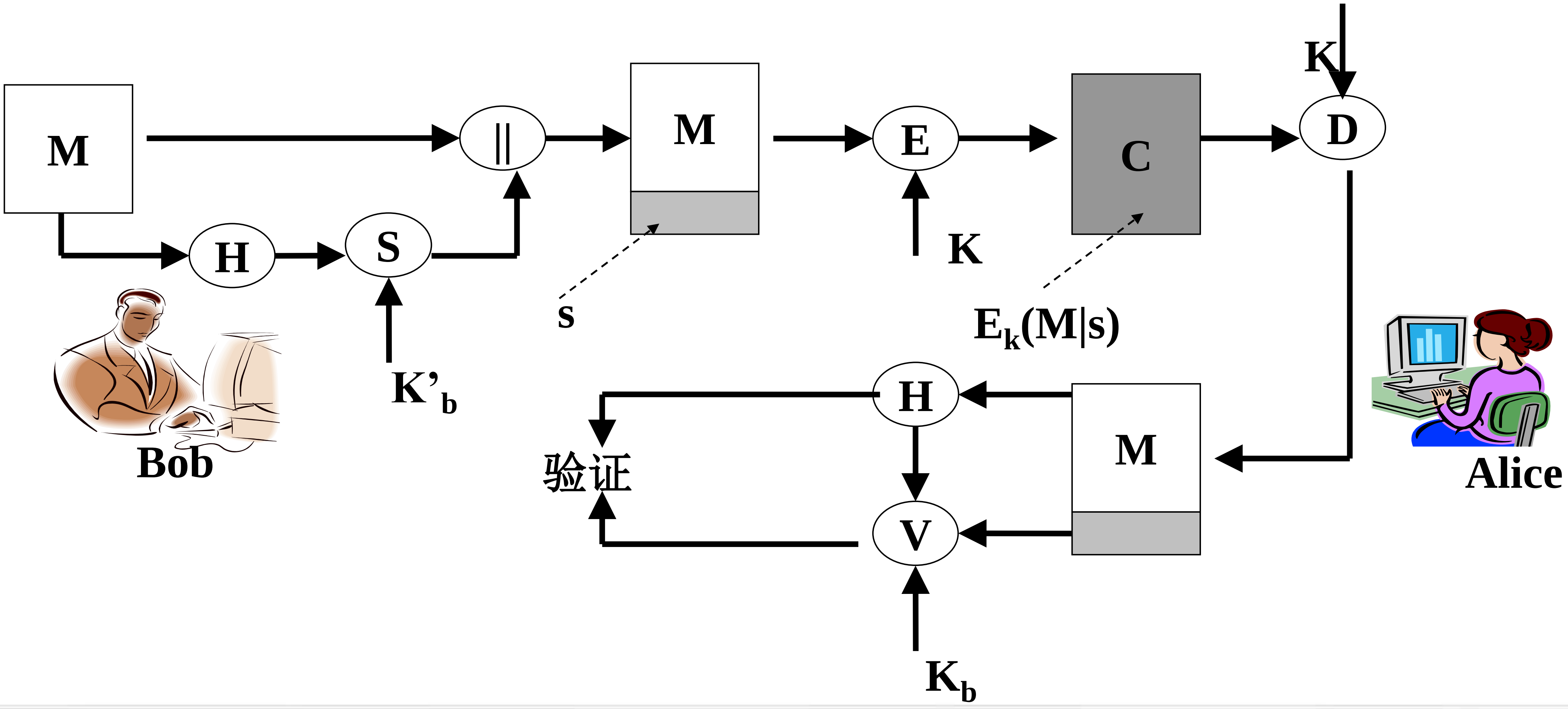
$ver(m,(r,s),y)=$真 $\Leftrightarrow$

$$g^{u_1} y^{u_2} \bmod p \bmod q = r$$

- 由于离散对数的困难性，敌手从**r**恢复**k**或从**s**恢复**x**都是不可行的
- 预计算： **r**的模指数运算$r=(g^k \bmod p) \bmod q$，这一运算与待签的消息无关。
- 用户可以预先计算出很多**r**和k^{-1}以备以后的签名使用，可大大加快签名的速度。

。

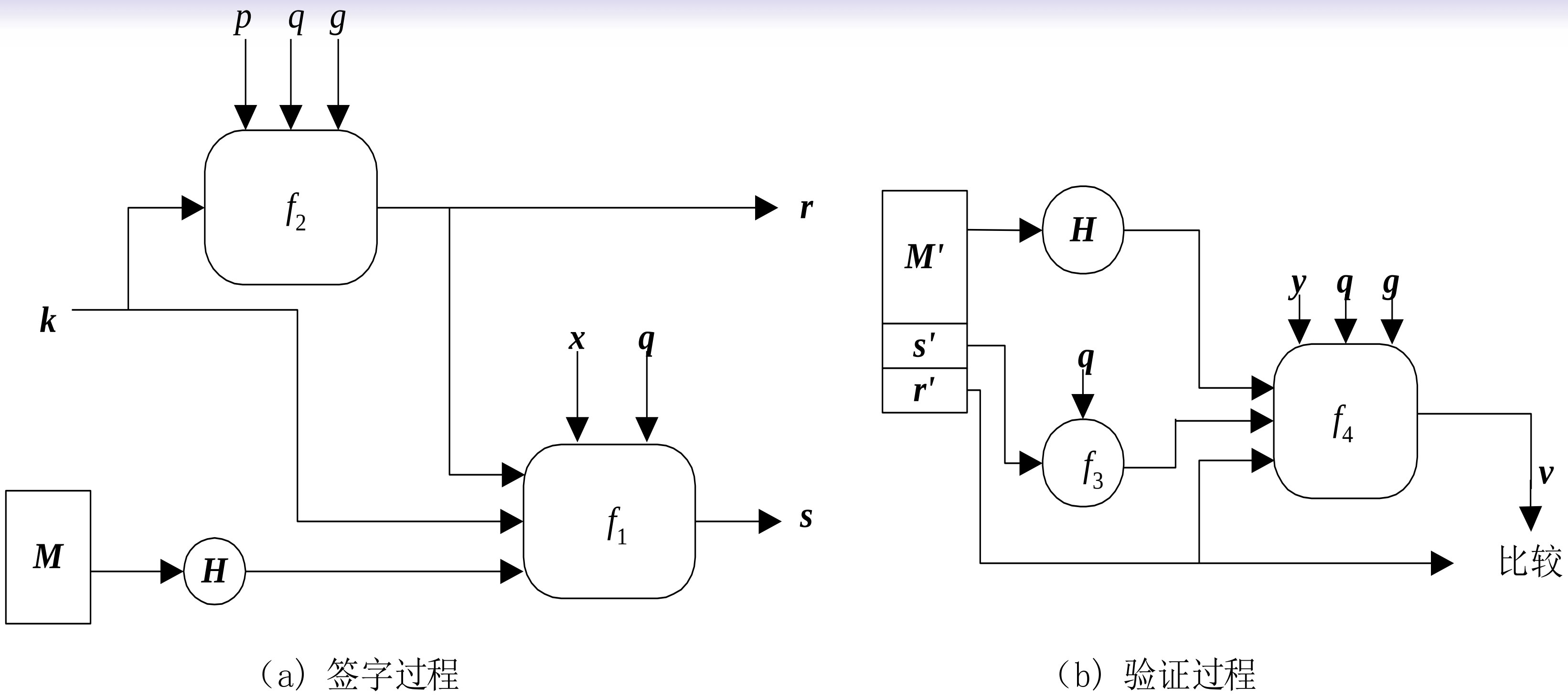
数字签名的一般用法



Hash函数在数字签名体制中的作用

- **A.** 破坏消息和签名之间的直接联系
- **B.** 提高签名效率
- **C.** 固定签名长度
- **D.** 增强签名体制的安全性
- **E.** 实现消息的完整性认证

DSS签名体制回顾



- 签名：

$$s \equiv f_1[H(M), k, x, r, q] \equiv [k^{-1}(H(M) + xr)] \bmod q$$

$$r = f_2(k, p, q, g) \equiv (g^k \bmod p) \bmod q$$

- 验证： $w = f_3(s', q) \equiv (s')^{-1} \bmod q$

$$v = f_4(y, q, g, H(M'), w, r') \equiv [(g^{(H(M')w) \bmod q} y^{r'w \bmod q}) \bmod p] \bmod q$$

6.3 其他签名方案

6.3.1 基于离散对数问题的数字签名体制

基于离散对数问题的数字签名体制是数字签名体制中最为常用的一类，其中包括

- ElGamal签名体制
- DSA签名体制
- Okamoto签名体制
- 。。。

1. 离散对数签名体制

ElGamal、DSA、Okamoto等签名体制都可归结为离散对数签名体制的特例。

(1) 体制参数

p：大素数；

q：p-1或p-1的大素因子；

g：$g\in_R Z^*_p$，且$g^q\equiv 1(\bmod p)$，其中$g\in_R Z^*_p$表示g是从Z^*_p中随机选取的，其中$Z^*_p=Z_p-\{0\}$；

x：用户A的私钥，$1<x<q$；

y：用户A的公钥，$y\equiv g^x(\bmod p)$。

(2) 签名:

$$s \equiv [k^{-1}(H(M)+xr)] mod\ q$$

对于待签名的消息m，A执行以下步骤：

① 计算m的杂凑值H(m)。

② 选择随机数k：1<k<q，计算$r \equiv g^k (mod\ p)$。

③ 从签名方程$ak \equiv b+cx(mod\ q)$中解出s。

$$\{a,b,c\} \leftrightarrow \{s,H(m),r\}$$

表6-1给出了这些可能选择中的一小部分，以(r, s)作为产生的数字签名。

(3) 签名验证:

接收方在收到消息m和签名(r, s)后，可以按照以下验证方程检验：

$$Ver(y,(r,s),m) = True \Leftrightarrow r^a \equiv g^b y^c (mod\ p)$$

$$ak \equiv b + cx \pmod q$$

表6-1 参数a,b,c可能的置换取值表

$\pm r'$	$\pm s$	$H(m)$
$\pm r'H(m)$	$\pm s$	1
$\pm r'H(m)$	$\pm H(m)s$	1
$\pm H(m)r'$	$\pm r's$	1
$\pm H(m)s$	$\pm r's$	1

$$Ver(y,(r,s),m) = True \Leftrightarrow r^a \equiv g^b y^c \pmod p$$

一些基于离散对数问题的签名方案

	签名方程	验证方程
①	$r'k \equiv s + h(m)x \bmod q$	$r^{r'} \equiv g^{s} y^{h(m)} \bmod p$
②	$r'k \equiv h(m) + sx \bmod q$	$r^{r'} \equiv g^{h(m)} y^{s} \bmod p$
③	$sk \equiv r' + h(m)x \bmod q$	$r^{s} \equiv g^{r'} y^{h(m)} \bmod p$
④	$sk \equiv h(m) + r'x \bmod q$	$r^{s} \equiv g^{h(m)} y^{r'} \bmod p$
⑤	$mk \equiv s + r'x \bmod q$	$r^{m} \equiv g^{s} y^{r'} \bmod p$
⑥	$mk \equiv r' + sx \bmod q$	$r^{m} \equiv g^{r'} y^{s} \bmod p$

$$Ver(y,(r,s),m) = True \Leftrightarrow r^{a} \equiv g^{b} y^{c} \pmod{p}$$

2. ElGamal签名体制

体制参数:

p: 大素数;

g: Z^*_p的一个生成元;

x: 用户A的秘密钥,

$x \in_R Z^*_p$;

y: 用户A的公开钥,

$y \equiv g^x (\bmod p)$。

签名：

用户为待签消息m选取

秘密随机数$k \in Z^*_{p-1}$

定义$Sig(m, k) = (r, s)$

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

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

验证：

$ver(m, (r, s), y) =$ 真 $\Leftrightarrow$

$$y^r r^s \equiv g^{H(m)} \bmod p$$

讨论两个问题：

（1）用**EIGamal**方案计算一个签名时，使用的随机数**k**能不能泄露？

（2）若**Bob**用相同的**k**值来签名不同的两份消息，**Oscar**能否攻破这个体制？

由于私钥 x 是保密的，所以，攻击者要得到这个密钥，必须求解离散对数问题 $\log_g y$ ，这是一个困难问题。但是，秘密随机数 k 一旦暴露，则解： $x = (h(m) - sk)r^{-1} \bmod (p-1)$ ，即可求得密钥 x，方案被攻破。

不可用同一个k作两次签名。若Bob利用相同k签名两次，即 m_1, m_2 的签名为 (r, s_2) 及 (r, s_1)，则Oscar可以由联立方程式

$$\begin{cases} h(m_1) = xr + ks_1 \bmod (p-1) \\ h(m_2) = xr + ks_2 \bmod (p-1) \end{cases}$$

可得：$x = [h(m_1)s_2 - h(m_2)s_1](s_2 - s_1)^{-1} r^{-1} \bmod (p-1)$，同时可以求得k。因此，同一个k不可重复使用。

3. Schnorr签名体制

体制参数:

p：大素数，$p \geq 2^{512}$；

q：大素数，$q|(p-1)$，$q \geq 2^{160}$；

g：$g \in_R Z^*_p$,且$g^q \equiv 1 (\bmod p)$；

x：A的秘密钥，$1<x<q$；

y：A的公开钥，$y \equiv g^x (\bmod p)$。

签名：

用户为待签消息m选取

秘密随机数$1 < k < q$

定义$Sig(m, k) = (e, s)$

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

$$e = H(r, m)$$

$$s = xe + k \bmod q$$

验证：

$$ver(m, (e, s), y) = 真 \Leftrightarrow$$

$$H(r, m) = e$$

$$r = g^k = g^s g^{-xe}$$

$$= g^s y^{-e} \bmod p$$

5. Okamoto签名体制

体制参数:

p：大素数，且p≥2^{512};

q：大素数，q|(p-1)，且q≥2^{140};

g_1, g_2：两个阶为q的元素, $g_1, g_2 \in Z_p^*$

x_1, x_2：用户A的秘密钥，两个小于q的随机数;

y：用户A的公开钥, $y = g_1^{-x_1} g_2^{-x_2} \bmod p$。

Okamoto签名体制

签名：

待签消息m,

选$k_1, k_2 \in_R Z_q^*$

定义$Sig(m, k_1, k_2) = (e, s_1, s_2)$

$$\begin{cases} e = H(g_1^{k_1} g_2^{k_2} \bmod p, m) \\ s_1 = k_1 + ex_1 \bmod q \\ s_2 = k_2 + ex_2 \bmod q \end{cases}$$

验证：

$ver(y, (e, s_1, s_2), m) =$ 真 $\Leftrightarrow$

$H(g_1^{s_1} g_2^{s_2} y^e \bmod p, m) = e$

DSS -----------> ECDSA

- (1) 全局公开钥
- p： $2^{L-1}<p<2^{L}$ 的大素数，$512 \leq L \leq 1024$且L是64的倍数。
- q： p-1的素因子，满足$2^{159}<q<2^{160}$ 。
- g： $g \equiv h^{(p-1)/q} \bmod p$，h满足$1<h<p-1$且使得$h^{(p-1)/q} \bmod p>1$的任一整数。
- (2) 用户秘密钥x：
- $0<x<q$的随机数。
- (3) 用户的公开钥y：
- $y \equiv g^{x} \bmod p$。

DSS ----------> ECDSA

签名:

用户为待签消息m选取秘密随机数$k, 0<k<q$

定义$Sig(m,k)=(r,s)$

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

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

$H(m)$:由SHA求出.

验证:

$$u_1=H(m)s^{-1} \bmod q$$

$$u_2=rs^{-1} \bmod q$$

$$ver(m,(r,s),y)=\text{真} \Leftrightarrow$$

$$g^{u_1}y^{u_2} \bmod p \bmod q=r$$

6.4 SM2椭圆曲线公钥密码签名算法

1 基本参数

与SM2椭圆曲线公钥密码加密算法的参数设置相同。

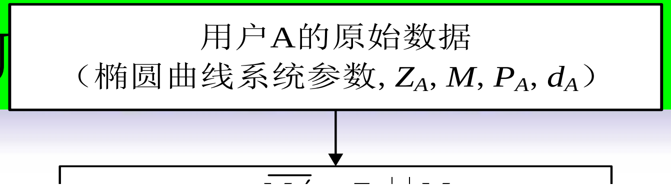
2 密钥产生

设签名方是A，A的秘密钥/公开钥的产生方式与SM2椭圆曲线公钥密码加密算法接收方B的产生方式相同，分别记为d_A和$P_A=(x_A,y_A)$。

基于素数域F_p的SM2算法参数如下：

- F_p的特征p，为m比特长的素数p，要尽可能大，但太大会影响计算速度；通常选择160比特大小。
- 长度不小于192比特的比特串$SEED$；
  - F_p上的2个元素a,b，满足$4a^3+27b^2\neq 0$，定义
  $$E(F_p): y^2=x^3+ax+b(modp)$$
- 基点$G=(x_G,y_G)\in E(F_p), G\neq O$；
- G的阶n为m比特长的素数，满足$n>2^{191}$且$n>4\sqrt{p}$
- $h=\frac{|E(F_p)|}{n}$称为余因子，其中$|E(F_p)|$是曲线$E(F_p)$的点数。

设ID_A是A的长度为$entlen_A$比特的标识，$ENTL_A$是由$entlen_A$转换而成的两个字节，A计算$Z_A = H_{256}\left(ENTL_A \| ID_A \| a \| b \| x_G \| y_G \| x_A \| y_A\right)$，其中*a,b*是椭圆曲线方程的参数、$(x_G, y_G)$是基点*G*的坐标，$x_A, y_A$是$P_A$的坐标。这些值转换为比特串后，再用$H_{256}$。验证 B验证签名时，也需计算$Z_A$。



3 签名算法

设待签名的消息为*M*，A做以下运算：

①取 $\bar{M}=Z_A\|M$ ；

②计算 $e=H_v(\bar{M})$，将 e 转换为整数，H_v 是输出为 v 比特长的哈希函数；

③用随机数发生器产生随机数

$$k\leftarrow_R\{1,2,\cdots,n-1\}$$

④计算椭圆曲线点 $C_1=kG=(x_1,y_1)$ ；

⑤计算 $r=(e+x_1)\bmod n$，若***r*=0**或 $r+k=n$ 则返回③；

⑥计算 $s=((1+d_A)^{-1}\cdot(k-r\cdot d_A))\bmod n$

若***s*=0**则返回③；

⑦消息**M**的签名为 (r,s) 。

1. 随机选择一个整数***k*, 1<*k*<*n***, 计算: ***kG*=(*x*, *y*), *r*=*x* mod *n***;
2. 计算***e*=*h*(*m*);**
3. 计算***s* = *k***

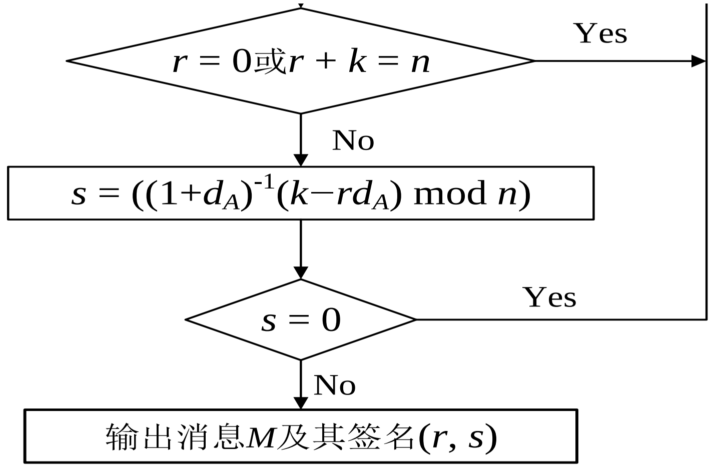
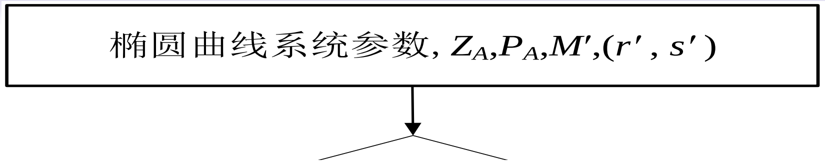
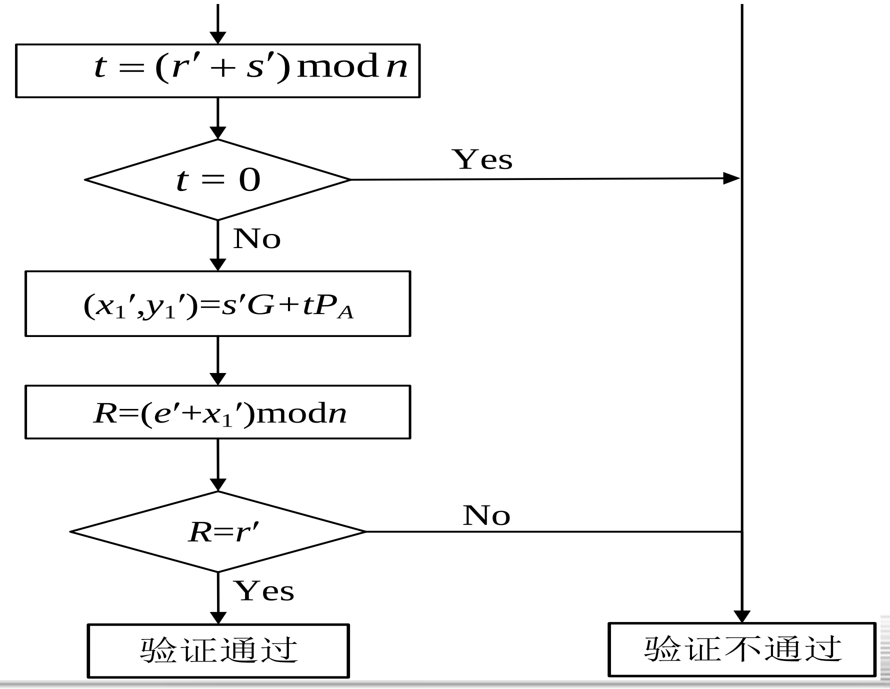


图6-4 SM2签名算法流程图

图6-5 SM2签名验证算法流程图



1. 计算$e=h(m)$;
2. $u=s^{-1}e, v=s^{-1}r$;
3. $(x_1,y_1)=uG+vP_A$, $r_1=x_1 \bmod n$;
4. 如果$r=r_1$, 则接受签名



4 验证算法

B收到消息M'及其签名(r',s')后，执行以下验证运算：

①检验$r' \in [1, n-1]$是否成立，若不成立则验证不通过；

②检验$s' \in [1, n-1]$是否成立，若不成立则验证不通过；

③置$\bar{M}' = Z_A \| M'$；

④计算$e' = H_v(\bar{M}')$，将e'转换为整数；

⑤计算$t = (r'+s') \bmod n$，若$t=0$；则验证不通过；

⑥计算椭圆曲线点$(x_1', y_1') = s'G + tP_A$；

⑦计算$R = (e' + x_1') \bmod n$，检验$R = r'$是否成立，若成立则验证通过；否则验证不通过。

正确性：如果 $\bar{M}'=\bar{M},(r',s')=(r,s)$**，则** $e'=e$ **，要证** $C_1=kG=(x_1,y_1)$ $R=r'=r$ **, 只需证** $x_1'=x_1$ **，因此需证明** $C_1=s'G+tP_A$**。**

$$s'G+tP_A=s'G+(r'+s')P_A=s'G+(r'+s')d_AG$$
$$=(s'+r'd_A+s'd_A)G=(s'(1+d_A)+r'd_A)G$$
$$=(k-r'd_A+r'd_A)G=kG$$

所以有 $x_1'=x_1$

6.5 身份证明技术

- 在很多情况下，用户都需证明自己的身份，如登录计算机系统、存取电子银行中的账目数据库、从自动出纳机ATM取款等。

  传统的方法：使用通行字或个人身份识别号PIN来证明自己的身份。

  缺点：检验用户通行字或PIN的人或系统可使用用户的通行字或PIN冒充用户。

- 身份证明技术，可使用户在不泄露自己的通行字或PIN的情况下向他人证实自己的身份。

交互证明系统

- 交互证明系统由两方参与

  证明者P （prover）

  验证者V （verifier）

- P知道某一秘密（如公钥密码体制的秘密钥或一平方剩余x的平方根），P希望使V相信自己的确掌握这一秘密。

- 交互证明比较典型的方式：每轮V都向P发出一询问，P向V做出一应答。

  所有轮执行完后，V根据P是否在每一轮对自己发出的询问都能正确应答，以决定是否接受P的证明。

- 交互证明和数学证明的区别： 数学证明的证明者可自己独立地完成证明，而交互证明是由P产生证明、V验证证明的有效性来实现，因此双方之间通过某种信道的通信是必需的。

- 交互证明系统须满足以下要求：

① 完备性：若P知道某一秘密，V将接受P的证明。

② 可靠性：如果P能以一定的概率使V相信P的证明，则P知道相应的秘密。

零知识证明

Alice: ``我知道密码学课的悬赏题的解答. "

Bob: ``你撒谎."

Alice: ``真的. "

Bob: ``不可能. "

Alice: ``千真万确. "

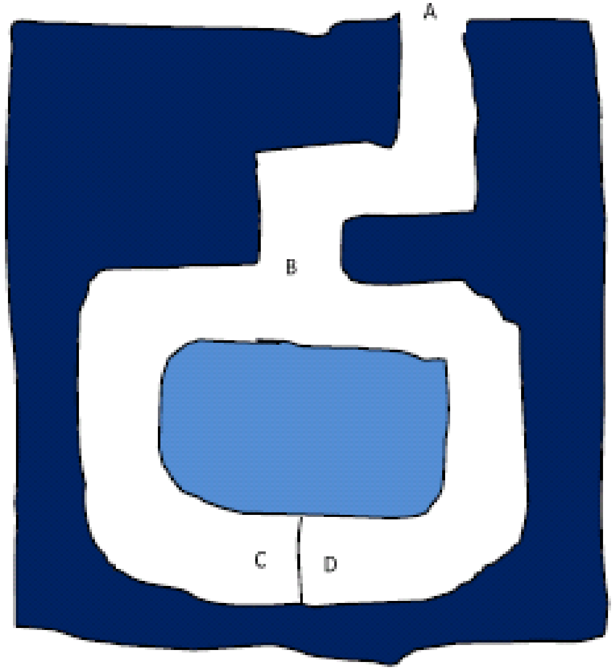
Bob: ``怎么证明? "

Alice: ``好吧,我告诉你!"

不好!

在这段对话中，**Alice**声称她能解决密码学课的悬赏题。她向**Bob**证明的方式是直接告诉**Bob**她的答案。这样一来，**Bob**也可以向老师领赏了。

零知识的例子

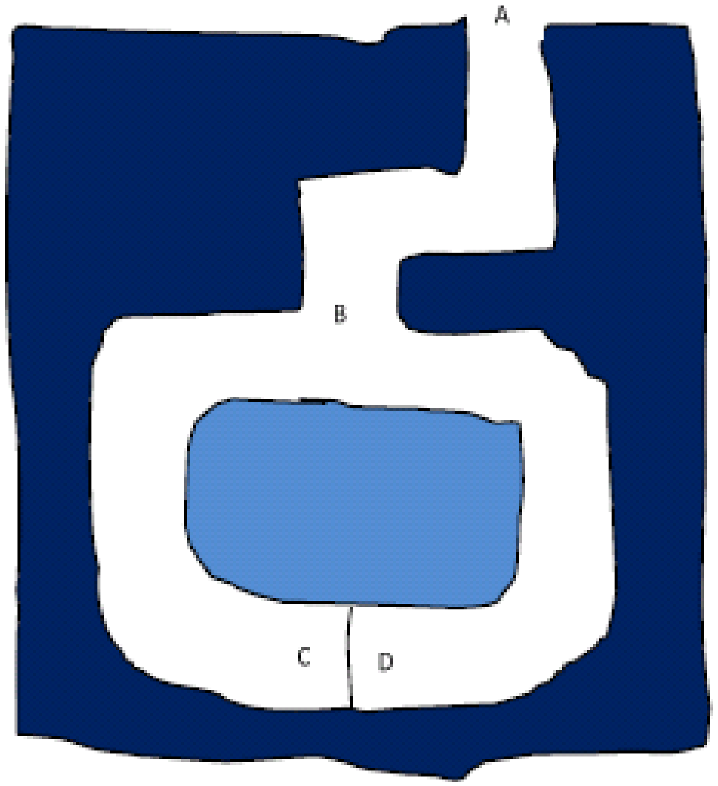


- 这是一个隧道. **C** 与 **D** 间有一道门，一般人是不能通过的.
- Alice 声称她能通过这道门.

•她不想让**Bob**知道她通过的诀窍，甚至也不希望泄露她到底能从哪个方向通过。

算法:

- ➢ **Alice**进入隧道，随机地沿着左边或者右边往下走。
- ➢ 等了一会儿，当**Bob**估计**Alice** 已经到达**C/D**门的时候,他来到位置**B**，随机大声叫**Bob**从左边或右边出来。
- ➢**Alice**听从**Bob**的指示沿着左边或右边出来。

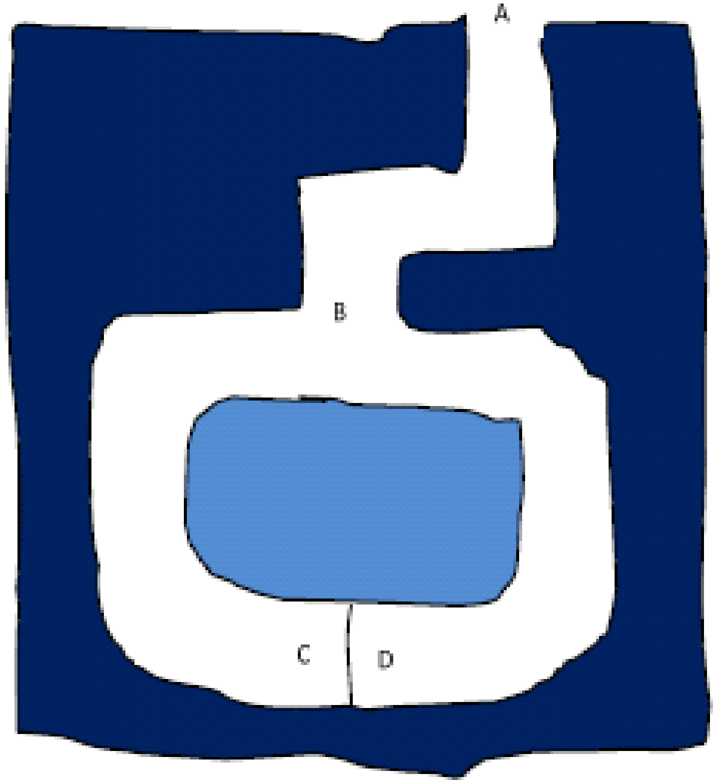


■ 重复以上试验很多次。如果**Alice**每一次都能成功地按**Bob**的指示出来，则**Bob**接受她的证明；否则，拒绝。

分析

➢不管Alice选择从哪一边进入隧道，都有50%概率与Bob指示出来的方向不一样。

➢如果进入方向与要求的出来方向不一样，则Alice必须使用她的能力才能通过C/D门。



➢如果她不知道通过门的方法，则她一次试验失败的概率是50%。 试验n次，她都成功的概率是2^{-n}.

➢Bob没有得到关于Alice怎样通过C/D门的任何信息。

零知识证明

- 在交互证明系统中，设**P**知道某一秘密，并向V证明自己掌握这一秘密，但又不向**V**泄露这一秘密，这就是最小泄露证明。

- 进一步，如果**V**除了知道**P**能证明某一事实外，不能得到其他任何信息，则称P实现了***零知识证明***，相应的协议称为零知识证明协议。

交互证明系统的定义

- 定义 6-1 我们称$(\mathcal{P}, \mathcal{V})$（或记为$\Sigma = \mathcal{P},\mathcal{V}$ ）是关于语言L、安全参数κ 的交互式证明系统，如果满足

  (1)完备性：$\forall x \in L$ ， $Pr[(P,V)[x]=1] \geq 1-\varepsilon(\kappa)$ ；

  (2)可靠性： $\forall \mathcal{P}^*$ ，$\forall x \notin L$， $Pr[(P^*,V)[x]=1] \leq \varepsilon(\kappa)$ ；

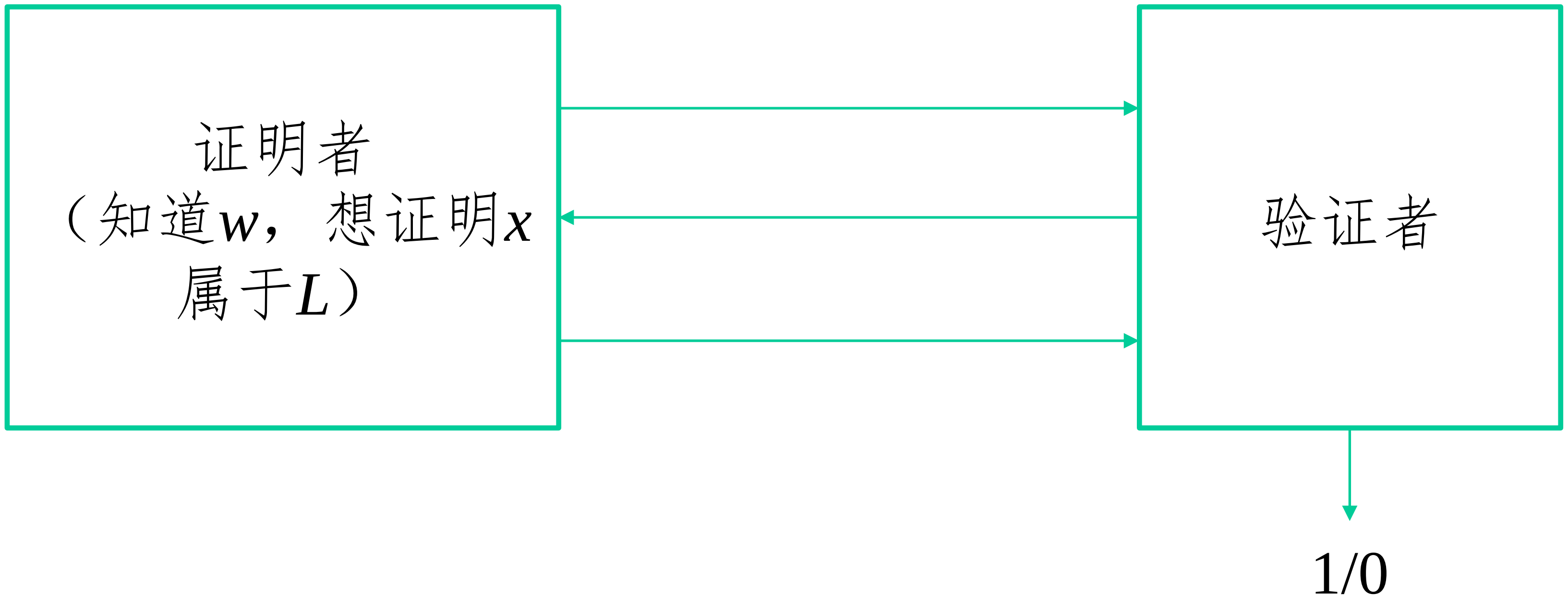
- 其中 $(\mathcal{P},\mathcal{V})[x]$ 表示当系统的输入是 x 时系统的输出。输出为1表示 $\mathcal{V}$接受 $\mathcal{P}$ 的证明。$\varepsilon(\kappa)$ 是可忽略的。

如何定义零知识证明

语言L：x属于L $\Longleftrightarrow$ 存在w使得$R(x,w)=1$

如何定义零知识证明

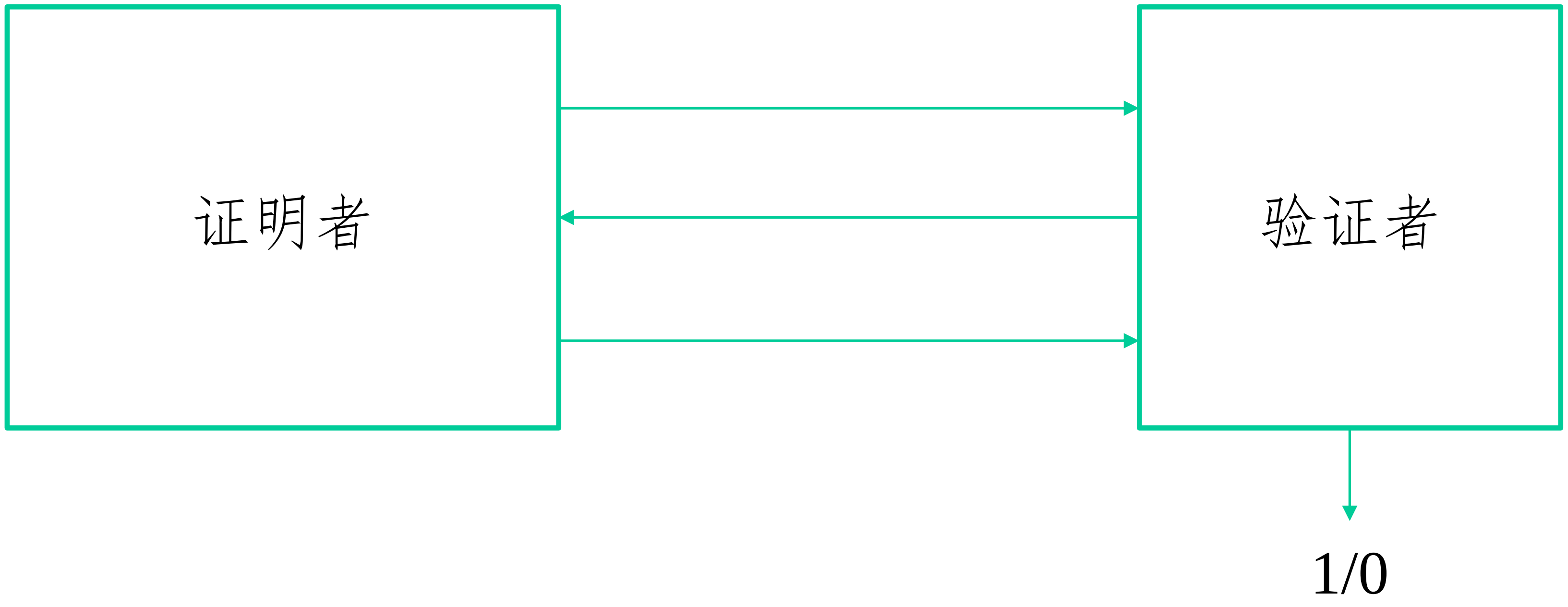
语言L：x属于L $\Longleftrightarrow$ 存在w使得$R(x,w)=1$



完备性：如果x属于L，验证者输出1（接受）的概率非常接近1。

如何定义零知识证明

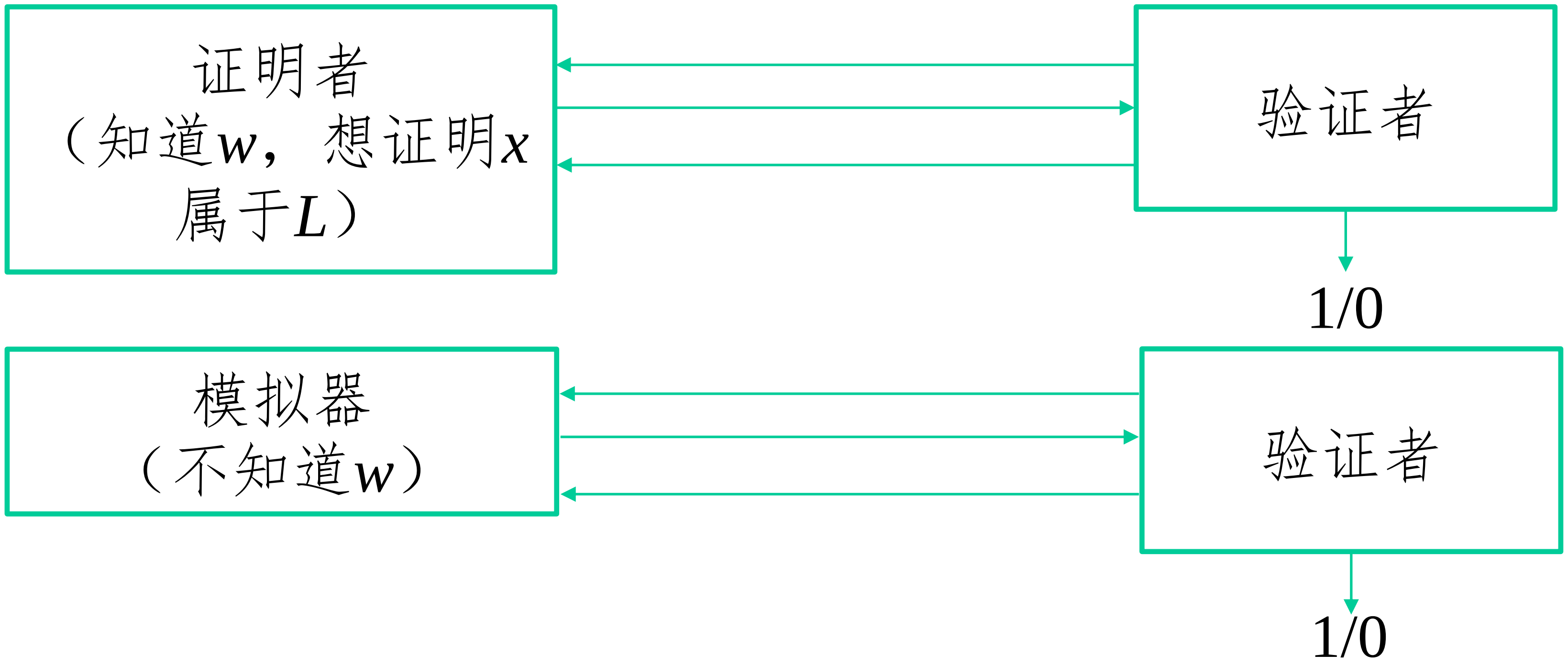
语言L：x属于L $\Longleftrightarrow$ 存在w使得$R(x,w)=1$



可靠性：如果x不属于L，验证者输出1（接受）的概率非常接近0。

如何定义零知识证明

语言L：x属于L $\Longleftrightarrow$ 存在w使得$R(x,w)=1$



零知识性：存在一个模拟器，可模拟验证者的视点（view）。

- 如何刻画交互式证明系统的零知识性，设交互式证明系统$\Sigma = (\mathcal{P},\mathcal{V})$用以证明$x \in L$。如果$\mathcal{V}$通过和$\mathcal{P}$交互得到的所有信息都能仅通过$x$计算得到，这就说明$\mathcal{V}$通过交互没有得到多余的信息。下面给出它的数学描述。
- 设$VIEW_{\mathcal{P},\mathcal{V}^*}(x)$是$\mathcal{V}^*$通过和$\mathcal{P}$交互（输入$x$）后得到的所有信息，包括从$\mathcal{P}$得到的消息和$\mathcal{V}^*$自己在协议执行期间选用的随机数，称为$\mathcal{V}^*$的view。
- 如果$VIEW_{\mathcal{P},\mathcal{V}^*}(x)$能在仅知道x的情况下，不通过交互而被模拟产生，则说明$\mathcal{V}^*$通过交互没有得到多余信息
用$\{VIEW_{\mathcal{P},\mathcal{V}^*}(x)\}_{x \in L}$表示$x \in L$时，$VIEW_{\mathcal{P},\mathcal{V}^*}(x)$的***概率分布***

定义6-2 设 $\Sigma = \mathcal{P},\mathcal{V}$ 是一交互证明系统，若对任意PPT的 $\mathcal{V}^*$，存在PPT的机器 S，使得对 $\forall x \in L$，$\{VIEW_{\mathcal{P},\mathcal{V}^*}(x)\}_{x\in L}$ 和 $\{S(x)\}_{x\in L}$ 服从相同的概率分布，记为 $\{VIEW_{\mathcal{P},\mathcal{V}^*}(x)\}_{x\in L} \equiv \{S(x)\}_{x\in L}$，则称 Σ 是**完备零知识的**，如果 $\{VIEW_{\mathcal{P},\mathcal{V}^*}(x)\}_{x\in L} \overset{c}{\equiv} \{S(x)\}_{x\in L}$，则称 Σ 是**计算上的零知识的**。

其中机器 S 称为模拟器，$S(x)$ 表示输入为 x 时，S 的输出；$\{S(x)\}_{x\in L}$ 表示 S 输出的概率。

简化的Fiat-Shamir身份识别方案

两个困难问题：

设n是一个大合数，找出n的所有素因子是一个困难问题，称之为大数分解问题。

给定一个合数n和整数a，判断a是否为$\bmod n$的二次剩余，这就是二次剩余问题。

定理 已知n为两个素数乘积, a是模n的平方剩余，则求解方程$x^2 \equiv a \bmod n$与分解n是等价的。

(1)已知n的分解n=pq，且a是模n的平方剩余，就可求得a mod n的4个平方根

$$x^2 \equiv a \bmod n \Leftrightarrow \begin{cases} x^2 \equiv a \bmod p \\ x^2 \equiv a \bmod q \end{cases} \Leftrightarrow \begin{cases} x \equiv \pm y \bmod p \\ x \equiv \pm z \bmod q \end{cases}$$

$$\begin{cases} x \equiv y \bmod p \\ x \equiv z \bmod q \end{cases} \quad \begin{cases} x \equiv y \bmod p \\ x \equiv -z \bmod q \end{cases} \quad \begin{cases} x \equiv -y \bmod p \\ x \equiv z \bmod q \end{cases} \quad \begin{cases} x \equiv -y \bmod p \\ x \equiv -z \bmod q \end{cases}$$

由中国剩余定理可求得a mod n的4个平方根，记为$\pm u \bmod n$和$\pm w \bmod n$，且$u \neq \pm w \bmod n$。

(2) 已知a mod n的两个不同的平方根（u mod n和w mod n，且u ≠±w mod n），就可分解n。

由$u^2 \equiv w^2 \bmod n$,得$(u+w)(u-w) \equiv 0 \bmod n$，但n不能整除u+w也不能整除u-w，所以必有

$$p|(u+w),\ q|(u-w)$$

或 $$p|(u-w),\ q|(u+w)$$

所以

$$\gcd(n, u+w)=p,\ \gcd(n, u-w)=q$$

或 $$\gcd(n, u-w)=p,\ \gcd(n, u+w)=q$$

因此得到了n的分解式。

简化的Fiat-Shamir身份识别方案

1. 协议及原理

设n=pq，p和q是两个不同的大素数，x是模n的平方剩余，y是x的平方根。n和x是公开的，而p、q和y是保密的。

y：证明者P的秘密。

证明者P和验证者V通过交互证明协议，P向V证明自己掌握秘密y，从而证明了自己的身份。

协议如下：

① P随机选r(0<r<n)，计算$a \equiv r^2 \bmod n$，将a发送给V。

② V随机选$e \in \{0,1\}$，将e发送给P。

③ P计算$b \equiv r y^e \bmod n$，即e=0时，b=r；e=1时，b=ry mod n。将b发送给V。

④ 若$b^2 \equiv a x^e \bmod n$，V接受P的证明。

在协议的前3步，P和V之间共交换了3个消息，这3个消息的作用分别是：

第1个消息是P用来声称自己知道a的平方根；

第2个消息e是V的询问，如果e=0，P必须展示a的平方根，即r，如果e=1，P必须展示被加密的秘密，即ry mod n；

第3个消息b是P对V询问的应答。

2. 协议的完备性、正确性和安全性

(1) 完备性

如果P和V遵守协议，且P知道y，则应答$b \equiv ry^e \bmod n$应是模n下ax^e的平方根在协议的第④步V接受P的证明，所以协议是**完备**的。

(2) 可靠性

假冒的证明者E可以1/2的概率骗得V接受自己的证明：

① E随机选r(0<r<n)和 $\tilde{e} \in \{0, 1\}$，计算 $a = r^2 x^{-\tilde{e}} \bmod n$ 将a发送给V。

② V随机选e∈{0,1}，将e发送给E。

③ E将r发送给V。

- 根据协议的第④步，**V**的验证方程$r^2 \equiv ax^e mod\ n \equiv r^2 x^{-\tilde{e}} x^e mod\ n$，当$\tilde{e} = e$时，验证方程成立，**V**接受**E**的证明，即**E**欺骗成功。因$\tilde{e} = e$的概率是**1/2**，所以**E**欺骗成功的概率是**1/2**。

- 根据协议的第④步，**V**的验证方程$r^2 \equiv ax^e mod\, n \equiv r^2 x^{-\tilde{e}} x^e mod\, n$，当$\tilde{e} = e$ 时，验证方程成立，**V**接受**E**的证明，即**E**欺骗成功。

  因$\tilde{e} = e$的概率是**1/2**，所以**E**欺骗成功的概率是**1/2**。

- 另一方面，**1/2**是**E**能成功欺骗的最好概率

- 根据协议的第④步，V的验证方程$r^2 \equiv ax^e \bmod n \equiv r^2 x^{-\tilde{e}} x^e \bmod n$，当$\tilde{e}=e$ 时，验证方程成立，V接受E的证明，即E欺骗成功。

  因$\tilde{e}=e$ 的概率是1/2，所以E欺骗成功的概率是1/2。

- 另一方面，1/2是E能成功欺骗的最好概率

否则假设E以大于1/2的概率使V相信自己的证明，那么E知道一个a，对这个a他可正确地应答V的两个询问e=0和e=1，意味着E能计算$b_1^2 \equiv a \bmod n$和$b_2^2 \equiv ax \bmod n$，即 $\frac{b_2^{\,2}}{b_1^{\,2}} \equiv x \bmod n$，因此E由 $\frac{b_2}{b_1} \bmod n$ 即可求得x的平方根y，矛盾。

- 根据协议的第④步，V的验证方程$r^2 \equiv ax^e \mathrm{mod}\ n \equiv r^2 x^{-\tilde{e}} x^e \mathrm{mod}\ n$，当 $\tilde{e}=e$ 时，验证方程成立，V接受E的证明，即E欺骗成功。

  因$\tilde{e}=e$的概率是1/2，所以E欺骗成功的概率是1/2。

- 另一方面，1/2是E能成功欺骗的最好概率

  否则假设E以大于1/2的概率使V相信自己的证明，那么E知道一个a，对这个a他可正确地应答V的两个询问e=0和e=1，意味着E能计算$b^2_1 \equiv a$ mod n和$b^2_2 \equiv ax$ mod n，即 $\frac{b_2^{\ 2}}{b_1^{\ 2}} \equiv x \bmod n$，因此E由 $\frac{b_2}{b_1} \bmod n$ 即可求得x的平方根y，矛盾。

- 假冒的证明者E欺骗V成功的概率是1/2，对V来说，这个概率太大了。为减小这个概率，可将协议重复执行多次，设执行t次，则欺骗者欺骗成功的概率将减小到2^{-t}。

零知识性（ P的安全性）：

① S随机选r(0<r<n)和 $\tilde{e} \in \{0, 1\}$，计算 $a = r^2 x^{-\tilde{e}} \bmod$ 将a发送给V。

② V随机选e∈{0,1}，将e发送给S。若 $\tilde{e} = e$ 不成立，回溯后重新进行第二步。

③ S将r发送给V。

由于V无法识别自己是在于P对话还是S对话，且与S的对话不泄露关于y的信息，与P的对话同样不泄露关于y的信息，零知识性成立。

Fiat-Shamir身份识别方案

1. 协议及原理

在简化的Fiat-Shamir身份识别方案中，验证者V接受假冒的证明者证明的概率是1/2，为减小这个概率，将证明者的秘密改为由随机选择的t个平方根构成的一个向量 $y=(y_1,y_2,\cdots,y_t)$，

模数n和向量 $x=(y^2_1,y^2_2,\cdots,y^2_t)$ 是公开的，其中n仍是两个不相同的大素数的乘积。

协议：

① P随机选r(0<r<n)，计算$a \equiv r^2 \bmod n$，将a发送给V。

② V随机选$e=(e_1,e_2,\ldots,e_t)$，其中$e_i \in \{0,1\}(i=1,2,\ldots,t)$，将e发送给P。

③ P计算 $b \equiv r\prod_{i=1}^{t} y_i^{e_i} \bmod n$ ，将b发送给V。

④ 若 $b^2 \not\equiv a\prod_{i=1}^{t} x_i^{e_i} \bmod n$ ，V拒绝P的证明，协议停止。

⑤ P和V重复以上过程k次。

2. 协议的完备性、可靠性和安全性

(1) 完备性

若**P**和**V**遵守协议，则**V**接受**P**的证明。

(2) 可靠性

如果假冒者E欺骗V成功的概率大于2^{-kt}，意味着E知道a，能正确地应答V的两个询问：

e=(e_1,e_2,…,e_t)和e=(f_1,f_2, …,f_t) 。

E能计算两个不同的值：

$$b_1^{\ 2} \equiv a\prod_{i=1}^{t} x_i^{\ e_i} \bmod n,\ b_2^{\ 2} \equiv a\prod_{i=1}^{t} x_i^{\ f_i} \bmod n$$

$$\frac{b_2^{\ 2}}{b_1^{\ 2}} \equiv \prod_{i=1}^{t} x_i^{\ f_i - e_i} \bmod n$$

假冒的证明者只有能正确猜测V的每次询问，才可使V相信自己的证明，成功的概率是2^{-kt}。

E能求得$\prod_{i=1}^{t} x_i^{f_i - e_i} \bmod n$的平方根为$\dfrac{b_1}{b_2} \bmod n$，矛盾。

Fiat-Shamir签名体制

(1) 体制参数

n：$n=pq$，其中p和q是两个保密的大素数；

k：固定的正整数；

$y_1, y_2, \ldots, y_k$：用户A的公开钥，对任何$i(1\le i\le k)$，y_i都是模n的平方剩余；

$x_1, x_2, \ldots, x_k$：用户A的秘密钥，对任何$i(1\le i\le k)$，$x_i \equiv \sqrt{y_i^{-1}} \pmod n$。

签名：

对于待签名的消息m，A执行以下步骤：

① 随机选一正整数t。

② 随机选t个介于1和n之间的数$r_1, r_2,\ldots, r_t$，并对任何$j(1\le j\le t)$，计算 $R_j = r_j^2 \bmod n$ 。

③ 计算$H(m, R_1, R_2,\ldots, R_t)$，并依次取出$H(m, R_1, R_2,\ldots, R_t)$ 的前kt个比特值$b_{11},\ldots,b_{1t}, b_{21},\ldots, b_{2t}, \ldots, b_{k1},\ldots, b_{kt}$。

④计算 $s_j = r_j \prod_{i=1}^{k} x_i^{b_{ij}} \bmod n$,$1\le j \le t$ 。

以$((b_{11},\ldots,b_{1t}, b_{21},\ldots, b_{2t}, \ldots, b_{k1},\ldots, b_{kt}), (s_1,\ldots, s_t))$作为对$m$的签名。

验证:

收到消息**m**和签名$((b_{11},\ldots,b_{1t}, b_{21},\ldots, b_{2t}, \ldots, b_{k1},\ldots, b_{kt}), (s_1,\ldots, s_t))$后，用以下步骤来验证:

① 计算 $R_j = s_j^2 \prod_{i=1}^{k} y_i^{b_{ij}} \bmod n$ ，$1 \le j \le t$。

② 计算$H(m, R_1, R_2,\ldots, R_t)$ 。

③ 验证$b_{11},\ldots,b_{1t}, b_{21},\ldots, b_{2t}, \ldots, b_{k1},\ldots, b_{kt}$是否依次是$H(m, R_1, R_2,\ldots, R_t)$ 的前**kt**个比特。如果是，则以上签名是有效的。

Schnorr身份识别方案

Schnorr协议需要一个可信中心，记为TA.

TA将为协议选择下列一些参数：

（1）p及q是两个大素数，且q|(p-1).q至少140位，而p至少为512位。

（2）$g\in Z^{*}_{p}$为q阶元

（3）h是一输出为t位的单向函数，t为一个安全参数；

（4）公开密钥y和秘密密钥x，用作签名。

p,q,h及其公开密钥都公布。每位用户自己选定个人秘密密钥$x\in[1,q-1]$,且计算公开密钥$y=g^{-x} \bmod p$。

Schnorr身份识别方案

证明者A能向验证者B证明他身份的协议（Schnorr识别协议）可描述为：

（1）用户A将其身份名I及公开密钥送交验证者B。验证者根据TA的数字签名来验证用户A的公开密钥。

（2）用户A任选一整数k，$1 \le k \le q-1$，计算$r=g^k \bmod p$，并将r送给验证者B。

（3）验证者B任选一整数$e \in [1,2^t]$，送给用户A。

（4）用户A送给验证者B：$s=k+xe \bmod q$；

（5）验证者B验证$r=g^s \times y^e \bmod p$.

3. Schnorr签名体制

体制参数:

p：大素数，$p \geq 2^{512}$；

q：大素数，$q|(p-1)$，$q \geq 2^{160}$；

g：$g \in_R Z^*_p$, 且 $g^q \equiv 1 (\mathrm{mod}\ p)$；

x：A的秘密钥，$1<x<q$；

y：A的公开钥，$y \equiv g^x (\mathrm{mod}\ p)$。

签名：

用户为待签消息m 选取

秘密随机数$1 < k < q$

定义$Sig(m, k) = (e, s)$

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

$$e = H(r, m)$$

$$s = xe + k \bmod q$$

验证：

$$\mathrm{ver}(m, (e, s), y) = \text{真} \Leftrightarrow$$

$$H(r', m) = e, r' \equiv g^s y^e \bmod q$$

$$r = g^k = g^s g^{-xe}$$

$$= g^s y^e \bmod p$$

Okamoto身份识别方案

TA将为协议选择下列一些参数：

p及q是两个大素数，$\alpha 1, \alpha 2 \in Z_p$为q阶元,对系统的所有参加者包括A，TA保密$c=\log_{\alpha 1}\alpha_2$。假定任何人计算c都是不可能的。TA选择一个签名方案和一个Hash函数。

Okamoto身份识别方案

TA向A颁布一个证书的协议为：

（1） TA建立A的身份并颁布一个识别串ID(A);

（2）A秘密地选择两个随机指数m1,m2, 1≤ m1,m2 ≤q-1，并计算$v=\alpha_1^{-m_1}\alpha_2^{-m_2} \bmod p$，将v发送给TA;

（3）TA对（ID,v）签名，$s=Sig_{TA}(ID,v)$。TA将证书C(A)=(ID(A),v,s)发送给A。

Okamoto身份识别方案

（1）A随机选择两个数r_1,r_2, $0\le r_1,r_2\le q-1$，并计算$X= \alpha_1^{r_1}\alpha_2^{r_2} \bmod p$;

（2）A将他的证书C(A)=(ID(A),v,s)和X发送给B;

（3）B通过检测$Ver_{TA}(ID,v)$=TURE来验证TA的签名。

（4）B随机选择一个数e,$1\le e\le 2^t$,t为安全参数并将e发送给A。

（5）A计算$y_1=(r_1+m_1e) \bmod q$,$y_2=(r_2+m_2e) \bmod q$，并将y_1, y_2发给B。

（6）B验证$X=\alpha_1^{y_1}\alpha_2^{y_2} v^e \bmod p$

6.6 一些特殊的签名

- 盲签名
- 盲签名方案是在发送者A和签名者B之间的双方协议。其基本思想如下。A发送给B一段信息，B对它签名并送回A。从这个签名，A能够计算B关于A预先所选消息m的签名。协议完成时，B既不知道消息m也不知道消息的签名。
- 盲签名的目的是防止B看到消息和签名，从而使B以后不能将所签消息和发送者A联系起来。

盲签名

- 盲签名的应用
  - 发送者A（客户）不希望签名者B（银行）能够将一条先验消息m及其签名$S_B(m)$与协议的特定实例相联系。
  - 这个特性在电子现金应用中可能很重要，因为那里的消息也许表示A所花的金钱数额。
  - 当m和$S_B(m)$提交给B进行支付时，B无法推断原先接收签名的是谁。这就是允许A的匿名性，从而A的消费模式不能被监测。

盲签名

- 盲签名协议需要下列组件：
  - 签名者B的一种数字签名机制。用$S_B(x)$记B对x的签名。
  - 函数f和g（只有发送者知道），满足$g(S_B(f(m))) = S_B(m)$。f叫做盲化函数，g叫做去盲函数，f(m)叫做盲消息。
- 第2条对S_B和g的选择加了许多限制。

盲签名

- **Chaum盲签名协议**
- 摘要：发送者A接收B关于盲消息的签名。由此A计算B关于A预先所选消息m的签名, $0 \le m \le n-1$。B既没有消息m也没有m相关签名的知识。
  - 记号。B的RSA公钥和私钥分别是(n, e)和d 。k是A随机选择的秘密数，满足$0 \le k \le n-1$且gcd (n, k)=1。
  - 协议步骤。
    - （盲化）A计算$m^* = mk^e \bmod n$, 将它发送给B 。
    - （签名）B计算$s^* = (m^*)^d \bmod n$, 将它发送给A。
    - （去盲）A计算$s = k^{-1} s^* \bmod n$，它就是B关于m的签名。

不可追踪电子现金

- **Chaum, Fiat,** 和**Naor**提出一个不可追踪电子现金方案。
  - 加入者**A**获取来自银行的电子现金货币
  - **A**可以将此货币在商店**B**花掉，而**B**无需与银行在线验证货币的真实性。
  - 当**B**在银行将电子货币兑现时，银行不能将其与**A**联系起来。
  - 如果**A**企图以该货币花两次（重复消费），那么**A**的身份就会暴露。
- **Okamoto**提出一种可分电子现金方案。可分电子硬币是一个与金钱数值关联的元素，可用来进行多次电子买卖，前提是所有交易的金钱总额不超过硬币数额。

不可否认的数字签名

- 不可否认的数字签名由Chaum和van Antwerpen 在1989年提出。
- 不可否认签名没有签名者的合作，接收者无法验证签名

不可否认签名的应用

- 实体A（客户）希望访问被实体B（银行）控制的某个安全区域。比如该安全区域可能是存放保险箱的房间。在许可访问之前，B要求A签署一份时间和日期的文件。如果A采用了不可否认签名，那么在验证过程中没有A的直接参与，（在以后的日期）B就不能向任何人证明A使用过安全区域中的设施。
- 假定某大公司A制作了一个软件包。A对软件包签名并将它卖给实体B，而B决定将其拷贝再卖给第三方C，那么没有A的合作C就无法验证该软件是否正版。当然，这种措施并不能阻止B用它自己的签名重新签署软件包，但因此B也就无法利用与A名气相关的市场利益。而且追踪B的欺诈行为也将很容易。

不可否认签名

- 一个不可否认签名方案有三部分组成：
  - 签名算法
  - 验证协议
  - 否认协议

不可否认签名

- 签名者可以声称一个签名是伪造的，在这种情况下，如果签名者拒绝参加验证，就可认为签名者有欺骗行为。如果签名者参加验证，由否认协议就可推断出签名的真伪。
- 否认协议需要做到以下两点
  - B能使A相信一个不合法的签名是伪造的。
  - B以很小的概率使A相信一个合法签名是伪造的。

不可否认签名

- 不可否认签名的一个不足之处是签名者有可能不在场或者拒绝合作，而导致签名无法被接收者验证。
  - Chaum提出“指定验证者签名”的概念，其中签名者指定某实体作为签名的验证者。
    - 一旦签名者不在场或者拒绝合作，验证者就有权力与接收者交互来检查签名。
    - 验证者不能产生签名者的签名。

不可否认的签名

（**Chaum**和**Van Antwerprn 1989**年提出）

该签名的特征是：验证签名者必须与签名者合作。验证签名是通过询问------应答协议来完成。这个协议可防止签名者**Bob**否认他以前做的签名。

不可否认的签名

（**Chaum**和**Van Antwerprn 1989**年提出）

该签名的特征是：验证签名者必须与签名者合作。验证签名是通过询问------应答协议来完成。这个协议可防止签名者**Bob**否认他以前做的签名。

一个不可否认的签名方案有三个部分组成：

签名算法、验证协议、否认协议

设**p=2q+1**是一个素数，它满足**q**为素数，且F_p中的对数问题是难解的。$\alpha \in F_p^*$,且阶为**q**,取**1≤a≤q-1**,定义 $\beta = \alpha^a (\bmod p)$ **G**表示阶为**q**的F_p^*的子群。易见$G=\langle\alpha\rangle$

设**P=A=G**，且定义$K=\{(p,\alpha,a,\beta) \mid \beta=\alpha^a(\bmod p)\}$,值$p,\alpha$和$\beta$是公开的，$a$是保密的。

对**K=（p,α,a,β)**和消息**x∈G,**定义**y=Sig$_K$(x)=x^a(modp)**

易见**y ∈G** 。按如下协议完成验证：

（1）**Alice** 随机选择 $e_1, e_2 \in F_q^*$

（2）**Alice**计算$C = y^{e_1}\beta^{e_2} \pmod p$，且将**C**送给**Bob.**

（3）**Bob**计算 $d = C^{a^{-1}(\text{mod}\, q)} \pmod p$ ，且**d**将送给**Alice.**

（4）**Alice**接受**y**作为一个有效签名**,**当且仅当

$$d \equiv x^{e_1}\alpha^{e_2} \pmod p$$

对上述这个签名方案，要证明以下两点：

1）**Alice**将会接受按如上方案的有效签名

2）**Bob**几乎不可否认经**Alice** 验证过的自己的签名。

证明(1)：**(Alice接受Bob的签名)**。下面计算的所有指数都已做到模**q**约简.

$$d \equiv C^{a^{-1}} (\bmod p) \equiv y^{e_1 a^{-1}} \beta^{e_2 a^{-1}} (\bmod p)$$

知 $y = x^a (\bmod p), \beta = \alpha^a (\bmod p)$

代入上式得 $d \equiv x^{e_1} \alpha^{e_2} (\bmod p)$

刚好与协议（**4**）相符，故**Alice**接受**Bob**的签名。

证明(1)：(**Alice**接受**Bob**的签名)。下面计算的所有指数都已做到模q约简.

$$d \equiv C^{a^{-1}} (\mathrm{mod}\ p) \equiv y^{e_1 a^{-1}} \beta^{e_2 a^{-1}} (\mathrm{mod}\ p)$$

知 $y = x^a (\mathrm{mod}\ p), \beta = \alpha^a (\mathrm{mod}\ p)$

代入上式得 $d \equiv x^{e_1} \alpha^{e_2} (\mathrm{mod}\ p)$

刚好与协议（4）相符，故**Alice**接受**Bob**的签名。

对于2）**Bob**几乎不可否认经**Alice**验证过的自己的签名。相当于证明下述定理。

定理1：若 $y \neq x^a \pmod p$,**那么Alice以概率1/q接受y作为x的有效签名.**

证明：**Bob对x做了签名y (=x^a)给Alice后。**

Bob接受了Alice的一个询问 $C = y^{e_1}\beta^{e_2} \pmod p$,**这个询问对应于q个有序对(e$_1$,e$_2$)。(原因是** $y, \beta \in G, C$ **一旦固定，e$_2$=f(e$_1$))**

然而，Bob不知Alice选择了哪一对(e$_1$,e$_2$)来构造出C。

- 如果$y \neq x^{a} \pmod{p}$，那么**Bob**能做的任何可能回答
- $d(=C^{a^{-1}(\bmod q)} \pmod{p}) \in G$ ， 刚好与**q-1**个可能的有序对**(e_1,e_2)**中的一个相对应。
- 由 **G=<α>**，所以对**c,d,x,y**来说，可设
- ***C*= α ^{i}, d= α^{j}, x= α^{k}, y= α^{l}, i, j, k, l** $\in F_q^*$ ，
- 考虑同余式：

$$C \equiv y^{e_1}\beta^{e_2} \pmod{p}, d \equiv (y^{e_1}\beta^{e_2})^{a^{-1}(\bmod q)} \pmod{p} \Rightarrow$$

$$d \equiv x^{e_1}\alpha^{e_2} \pmod{p}$$

- 写出关于 α 的指数表示：

$$\begin{cases} \alpha^{i} \equiv \alpha^{le_1} \cdot \alpha^{ae_2} \pmod{p} \\ \alpha^{j} \equiv \alpha^{ke_1} \cdot \alpha^{e_2} \pmod{p} \end{cases}$$

- 等价于下述方程组：

$$\begin{cases} i = le_1 + ae_2 \pmod{q} \\ j = ke_1 + e_2 \pmod{q} \end{cases}$$

既然假设 $y \neq x^a \pmod{p}$ 而 **y= α^{l}，x^{a}=(α^{k})^{a}= α^{ak}，**所以 **l≠ak,**

相当于说上述方程的系数行列式：

$$\begin{vmatrix} l & a \\ k & 1 \end{vmatrix} = l - ak \neq 0 \pmod{q}$$

知该方程组仅有唯一一组解。

即对每一个**d∈G,** 对于**q**个可能的有序对中（**e$_1$,e$_2$**），刚好有一个是正确的回答，**Bob**给**Alice**的一个回答**d**，将被验证通过的概率刚好为**1/q**。定理得证！

下面讨论否认协议：

目的：（1）**Bob**能使**Alice**相信一个无效的签名是伪造的.

（2）**Bob**签名有效，而导致**Alice**判决错误的概率为小概率事件。

否认协议：**(y?=x^a)**暂视为对的签名

1）**Alice** 随机选取 $e_1, e_2 \in F_q^*$

2）**Alice**计算 $C = y^{e_1}\beta^{e_2} \pmod p$ 且发送给**Bob,**

3）**Bob**计算 $d = C^{a^{-1} \pmod q} \pmod p$,且发送给**Alice**

4）**Alice**验证 $d \neq x^{e_1}\alpha^{e_2} \pmod p$

5）**Alice**再随机选取 $f_1, f_2 \in F_q^*$

6）**Alice**计算 $C' = y^{f_1}\beta^{f_2}(mod\ p)$，且发送给**Bob**

7）**Bob**计算 $D = C'^{a^{-1} \pmod q} \pmod p$，且发送给**Alice**

8）**Alice**验证 $D \neq x^{f_1}\alpha^{f_2} \pmod p$

9)Alice推出 y是伪造的

$$\Leftrightarrow (d\alpha^{-e_2})^{f_1} \equiv (D\alpha^{-f_2})^{e_1} (\bmod\ p)$$

定理2：如果 $y \neq x^a (\bmod p)$,且Alice和Bob都遵守否认协议，那么

$$(d\alpha^{-e_2})^{f_1} \equiv (D\alpha^{-f_2})^{e_1} (\bmod p)$$

证明：注意， $d \equiv C^{a^{-1}} (\bmod p)$， 而 $C \equiv y^{e_1}\beta^{e_2} (\bmod p)$

又 $\beta = \alpha^a (\bmod p)$ ，从而有

$$(d\alpha^{-e_2})^{f_1} \equiv ((y^{e_1}\beta^{e_2})^{a^{-1}} \alpha^{-e_2})^{f_1} (\bmod p)$$

进一步有

$$\begin{aligned}(d\alpha^{-e_2})^{f_1} &\equiv y^{e_1 a^{-1} f_1} \beta^{e_2 a^{-1} f_1} \alpha^{-e_2 f_1} (\bmod p) \\ &\equiv y^{e_1 a^{-1} f_1} \alpha^{a e_2 a^{-1} f_1} \alpha^{-e_2 f_1} (\bmod p) \\ &\equiv y^{e_1 a^{-1} f_1} \alpha^{e_2 f_1} \alpha^{-e_2 f_1} (\bmod p) \\ &\equiv y^{e_1 a^{-1} f_1} (\bmod p)\end{aligned}$$

类似地，按如上方式推出

$$(D\alpha^{-f_2})^{e_1} \equiv y^{e_1 a^{-1} f_1} \pmod p$$

证毕。

注：我们不能假设**Bob**遵守了否认协议，他可以想方设法构造**d,D,**来达到否认自己签过名的目的。然而，只要**Alice**严格遵守协议，**Bob**是无法否认的。可证。

定理**3**，假设**y≡x^a(modp)** 且**Alice**遵守否认协议，如果

$$d \neq x^{e_1}\alpha^{e_2} \pmod p$$

$$D \neq x^{f_1}\alpha^{f_2} \pmod p$$

那么 $(d\alpha^{-e_2})^{f_1} \neq (D\alpha^{-f_2})^{e_1} \pmod p$

成立的概率为**1-1/q**。

群签名

- **1991年，Chaum和Van Heijst提出群签名方案。**
- 该方案允许群众的某个成员以群的名义匿名地签发消息。满足下述三个条件：
  - 只有群中的成员才能代表群进行签名；
  - 签名的接收者能验证签名是哪一个群的一个合法签名，但不能分辨具体的签名者。
  - 一旦出现争端，可借助群成员或一个可信的机构能识别出签名者。

群签名的应用

- 一个公司有几台计算机，每台都联在局域网上。公司的每个部门有其自己的打印机（也连在局域网上），并且只有本部门的人员才能允许使用其部门的打印机。因此，打印前必须确认用户在哪个部门工作。同时公司为了保密，不可以暴露用户的身份。然而，如果有人滥用打印机，主管者必须能找出是谁在滥用打印机。

一个群签名方案由以下几个部分组成：

（1）建立（**setup**） 一个用以产生群公钥和私钥的多项式概率算法。

（2）加入（**join**） 一个用户和群管理员之间的交互式协议。执行该协议可以使用户成为群成员，群管理员得到群成员的秘密的成员管理密钥，并产生群成员的私钥和成员证书。

（3）签名（**sign**） 一个概率算法，当输入一个消息、一个群成员的私钥和一个群公钥后，输出对该消息的签名。

（4）验证（**verify**） 给定一个消息的签名和一个群公钥后，判断该签名相对于该群公钥是否有效。

（5）打开（**open**） 给定一个签名、群公钥和群私钥的条件下确定签名者的身份。

一个好的群签名方案应该满足以下几条性质：

① 正确性（correctness）

② 不可伪造性（unforgeability）

③ 匿名性（anonymity）

④ 不可关联性（unlinkability）

⑤ 可跟踪性（traceability）

⑥ 可开脱性（exculpability）

⑦ 抗联合攻击（coalition-resistance）

2. 代理签名

一个代理签名方案由以下几个部分组成：

- 系统建立　选定代理签名方案的系统参数，用户的密钥等。
- 签名权力的委托　原始签名者将自己的签名权力委托给代理签名者。
- 代理签名的产生　代理签名者代表原始签名者产生代理签名。
- 代理签名的验证　验证人验证代理签名的有效性。

2．代理签名

根据签名权力委托的方式不同，代理签名可以分为以下几类：

（1）完全代理（**full delegation**）

（2）部分代理（**partial delegation**）

（3）具有证书的代理（**delegation by warrant**）

（4）具有证书的部分代理（**partial delegation with warrant**）

2．代理签名

根据原始签名者能否产生同代理签名者一样的签名，代理签名又可分为两类：

1）代理非保护（**proxy-unprotected**） 原始签名者能够产生有效的代理签名。

2）代理保护（**proxy-protected**） 原始签名者不能够产生有效的代理签名。

一个强代理签名方案应满足以下几条性质：

① 可区分性（distinguishability）
② 可验证性（verifiability）
③ 强不可伪造性（strong unforgeability）
④强可识别性（strong identifiability）
⑤ 强不可否认性（strong undeniability）
⑥ 防止滥用（prevention of misuse）

谢谢！