第7章 数字签名

杨礼珍

同济大学计算机科学与技术系, 2018

Outline

本章作业

课本习题7.4、7.5(a)
思考题：课本习题7.2,7.3,7.5(b)
本章学习难点：对协议的安全性分析
本章学习重点掌握：签名方案基本要求、定义，签名方案与hash函数、加密的关系，RSA签名方案与安全性分析，Elgamal签名方案及安全性分析。

7.1 引言

协议定义：

- 包括一系列步骤
- 至少有2个参与方
- 目标是完成某项任务

密码协议：使用密码学的协议

- 参与协议的伙伴可能是朋友和完成信任的人，也可能是敌人或者互相不信任的人
- 包含某些密码算法
- 使用密码的目的是防止或者发现窃听者和欺骗。

将要学习的基本密码协议：

- 签名方案
- 认证方案
- 密钥分配方案
- 密钥协商方案

7.1 引言

签名方案

- 数字世界的问题：
  - Alice和Bob如何确认签订的合同？
  - Bob如何确认收到的邮件来自Alice？
  - Alice在网上购物，如何向银行确认支付订单？
  - ...
- 传统方法是采用手写签名，但存在问题：
  - 无法应用到数字世界
  - 手写签名容易被伪造
- 签名方案(signature scheme)是一种以电子形式存储的消息签名的方法，也称为数字签名(digital signature)
- 安全的数字签名需要解决以下基本问题
  - 签名与消息绑定(其它人无法伪造有效签名，签名者无法否认合法签名)
  - 其他人能够验证签名的有效性
  - 能够防止签名被重复使用：加入签名时间等信息解决

7.1 引言

签名方案

1. 签名算法$sig_K, K \in \mathcal{K}$
2. 验证算法$ver_K, K \in \mathcal{K}$
3. 签名：Alice使用她的私钥SK对消息x签名
$$y = sig_{SK}(x)$$
把(x, y)发送给Bob
4. 验证：Bob收到Alice的签名(x, y)后，使用Alice的公钥PK如下验证:
$$ver_{PK}(x, y) = \begin{cases} \text{true} & \text{认为} y = sig_{SK}(x) \\ \text{false} & \text{认为} y \neq sig_{SK}(x) \end{cases}$$

7.1 引言

Alice　　　　　　　　　　Bob

产生签名私钥SK和公钥PK

$y=sig_{SK}(x)$　　(x, y) →

获得Alice的签名公钥PK

验证签名：$Ver_{pK}(x, y)$

7.1 引言

签名和公钥加密使用的密钥对比

签名	签名验证	公钥加密	公钥解密
签名者私钥	签名者公钥	接收者公钥	接收者私钥

7.1 引言

密码体制7.1 RSA签名方案

- 公钥$PK=(n,b)$，其中$n=pq$，p和q为素数
- 私钥$SK=a$，满足：$ab\equiv 1 \bmod (p-1)(q-1)$
- 签名算法：对消息$x\in\mathbb{Z}_n$，签名如下计算

$$sig_{SK}(x)=x^a \bmod n$$

- 验证算法：对签名消息(x,y)如下验证

$$ver_{PK}(x,y)=\begin{cases} true & \text{for } x=y^b \bmod n \\ false & \text{for } x\neq y^b \bmod n \end{cases}$$

7.1 引言

RSA签名的有效性讨论：

- RSA签名的公钥私钥和加密算法的公钥私钥一样。
- 签名和验证算法和RSA的加解密一样
- 因为对应解密算法是有效的，合法签名可以通过验证算法
- 对于任何人，都可以使用Alice的公钥验证她的签名。

RSA签名方案是一个简单的签名算法，但是并不安全！

7.1 引言

RSA签名的安全问题例1：

1. 假定b, a分别为Alice的RSA加解密指数
2. Oscar选择任意签名y，计算消息$x = y^b \bmod n$（注意b是公开的）
3. Bob验证：$x = y^b \bmod n$，因此他认为(x, y)为Alice的签名，但实际上Alice没有对x签名。

7.1 引言

RSA签名的安全问题例2：

1. 假定a, b分别为Alice的RSA加解密指数
2. Alice发布对两条消息x_1, x_2的签名

$$y_1 = x_1^a \bmod n, y_2 = x_2^a \bmod n$$

3. Oscar可计算消息$x_1 x_2$的签名

$$sig_a(x_1 x_2) = y_1 y_2 \bmod n (= (x_1 x_2)^a \bmod n)$$

7.1 引言

RSA签名的安全问题例3：

1. 假定a, b分别为Alice的RSA加解密指数
2. Oscar为了伪造不利于Alice的消息x的签名，他找到两条消息x_1, x_2满足$x = x_1 x_2 \bmod n$
3. Oscar请求Alice对消息x_1, x_2的签名，得到

$$y_1 = x_1^a \bmod n, y_2 = x_2^a \bmod n$$

4. Oscar可计算消息x的签名

$$sig_a(x_1 x_2) = y_1 y_2 \bmod n (= (x_1 x_2)^a \bmod n)$$

Alice向Bob发送消息x，如果她想通过签名向Bob确认消息来源：

- (方案一（先签名后加密））：
  1. Alice先对消息签名得到$(x, y = sig_{SK}(x))$，然后使用Bob的公钥Bob对签名加密$z = e_{Bob}(x, y)$
  2. Bob接受到消息z后，先用他的私钥解密得到(x, y)，然后用Alice的公钥验证签名，从而确认消息来自Alice。
- (方案二（先加密后签名）， 不安全：
  1. Alice先加密消息$z = e_{Bob}(x)$，然后计算加密消息z的签名$y = sig_{SK}(z)$
  2. Oscar截获到Alice发送给Bob的消息(z, y)，对z计算他的签名：
  $$y' = sig_{Osacr}(z)$$
  然后用(z, y')代替(z, y)发送给Bob
  3. Bob使用Osacr的公钥验证(z, y')，Bob确认消息来自Osacr， 而不是Alice(例如，这是Alice对付款信息的签名，将面临经济损失)

7.1 引言

总结：一个看似安全的签名方案可能存在未知的安全隐患，需要对安全性进行深入分析

7.2 签名方案的安全性需求

根据敌手所掌握的信息，对签名方案的攻击分类：

- 唯密钥攻击：Oscar拥有Alice的公钥
- 已知消息攻击：Oscar拥有一系列Alice的签名消息，例如：$(x_1, y_1), (x_2, y_2) \ldots$，其中$y_i = sig_K(x_i)$
- 选择消息攻击：Oscar请求Alice对一系列消息$(x_1, x_2, \ldots)$签名，得到$y_i = sig_K(x_i), i = 1, 2 \ldots$

敌手的攻击目标分类:

- 完全破译：Oscar确定出Alice的私钥，这样可对任意消息伪造Alice的签名
- 选择性伪造：对Alice没有签名过的消息x，Oscar能够以某种概率计算出Alice的有效签名y，即$ver_K(x, y) = true$
- 存在性伪造：Oscar能够产生Alice的一对有效签名(x, y)，其中$ver_K(x, y) = true$，且x不是Alice签名过的消息。

7.2 签名方案的安全性需求

例子：

- RSA签名的安全问题例1是唯密钥攻击的存在性伪造。
- RSA签名的安全问题例2是已知消息攻击的存在性伪造。
- RSA签名的安全问题例3是选择消息攻击的选择性伪造。

7.2.1 签名和Hash函数

我们讨论过，在RSA签名方案中，如果知道消息x_1, x_2的签名y_1, y_2，那么$x_1 x_2$的签名为$y_1 y_2 \bmod n$。
为了抵抗以上攻击，可以采取以下措施：(即对消息的hash值签名，对消息的hash值验证，而不是对消息本身签名和验证)

- 签名方案：
  1. 计算消息x的Hash值$z = h(x)$
  2. 计算z的签名$y = sig_K(z)$
  3. 发送(x, y)
- 验证方案：计算$z = h(x)$，$ver_K(z, y)$。

一般数字签名采用以上方式工作。
第4章讨论过，hash和签名结合的方案中，hash函数需满足单向、第二原像稳固和碰撞稳固。

7.3 ElGamal 签名方案

- ElGamal签名方案发表于1985年，和ElGamal加密同为Taher ElGamal设计。
- ElGamal签名方案存在安全缺陷，在实际中应用很少，但其变形算法应用广泛，其变形算法有Schnorr签名方案，及采纳为标准的数字签名算法(DSA)。
- 学习难点：ElGamal的安全性分析

密码体制7.2 ElGamal签名方案

密钥生成(和ElGamal加密的密钥一样)：

- 公钥：(p, α, β)，其中p为素数，$\alpha \in \mathbb{Z}_p^*$是本原元，$\beta = \alpha^a \pmod p$
- 私钥：$a \in \mathbb{Z}_{p-1}^*$

签名算法：

1. 输入：消息$x \in \mathbb{Z}_p^*$
2. 产生秘密随机数$k \in \mathbb{Z}_{p-1}^*$
3. 对x的签名$sig_K(x, k) = (\gamma, \delta)$，其中

$$\gamma = \alpha^k \pmod p, \delta = (x - a\gamma)k^{-1} \pmod{p-1}$$

验证算法：

- 输入：(x, γ, δ)
- $ver_K(x, (\gamma, \delta)) = true \Leftrightarrow \beta^\gamma \gamma^\delta \equiv \alpha^x \pmod p$

在后面内容中应用广泛的两个引理

引理1：

对群$(G,\cdot)$上的元素α，如果正整数m满足$\alpha^m = 1$，如果有$x \equiv y \pmod{m}$，那么$\alpha^x = \alpha^y$。

证明：由$x \equiv y \pmod{m}$可令$x = y + km$，因此$\alpha^x = \alpha^{y+km} = \alpha^y \alpha^{km} = \alpha^y$。

引理2：

对群$(G,\cdot)$上的m阶元素α（即m是满足$\alpha^k = 1$的最小正整数），有：

$$\alpha^x = \alpha^y \text{ 等价于} x \equiv y \pmod{m}$$

证明：

1.必要性。 如有$\alpha^x = \alpha^y$，那么$\alpha^{x-y} = 1$。不妨设$x - y = km + r, 0 \le r < m$。有

$$\alpha^{km+r} = \alpha^r = 1$$

如果$r > 0$，又$r < m$，则与α是m阶元素的结论矛盾，因此$r = 0$，这意味着$x \equiv y \pmod{m}$。

2.充分性。由引理1即得。

根据以上结论，我们可以把指数运算转化为对数运算，或者把对数运算转化为指数运算。

对有效签名，ElGamal验证算法输出为true（应用了引理2）：

$$\delta = (x - a\gamma)k^{-1} \pmod{p-1} \quad \Leftrightarrow \quad x \equiv a\gamma + k\delta \pmod{p-1}$$

$$\Updownarrow$$

$$\alpha^x \equiv \alpha^{a\gamma}\alpha^{k\delta} \pmod{p}$$

$$\Updownarrow$$

$$ver_K(x, (\gamma, \delta)): \quad \alpha^x \equiv \beta^\gamma \gamma^\delta \pmod{p}$$

Example

例7.1 假定选取$p = 467, \alpha = 2, a = 127$，那么

$$\beta = \alpha^a \bmod p = 2^{127} \bmod 467 = 132$$

Alice如下对消息$x = 100$签名：

1. 选取随机数$k = 213$，
   且$k^{-1} \bmod p - 1 = 213^{-1} \bmod 466 = 431$
2. 计算：

$$\begin{aligned} \gamma &= 2^{213} \bmod 467 = 29 \\ \delta &= (100 - 127 \times 29)431 \bmod 466 = 51 \end{aligned}$$

任何人都可以验证签名：

$$132^{29} \times 29^{51} \equiv 189 \pmod{467}$$

ElGamal签名的唯密钥存在性伪造就是：

- 已知：p, α, β
- 构造出(x, γ, δ)满足:

$$\beta^{\gamma}\gamma^{\delta} \equiv \alpha^{x} \pmod{p} \tag{1}$$

令$\gamma = \alpha^{i}\beta^{j} \bmod p$，等式 (1)等价于

$$\beta^{\gamma}(\alpha^{i}\beta^{j})^{\delta} \equiv \alpha^{x} \pmod{p} \Leftrightarrow \alpha^{x-i\delta} \equiv \beta^{\gamma+j\delta} \pmod{p}$$

为令上式成立，可另两边指数为0，即

$$x - i\delta \equiv \gamma + j\delta \equiv 0 \pmod{p-1}$$

解得x, γ，所构造得到的(γ, δ)是消息x的有效签名：

$$\begin{aligned}
\gamma &= \alpha^{i}\beta^{j} \bmod p \\
\delta &= -\gamma j^{-1} \bmod p-1 \\
x &= -\gamma i j^{-1} \bmod p-1
\end{aligned}$$

Example

例7.2 设$p = 467, \alpha = 2, \beta = 132$。根据上面的构造方法，Oscar选择$i = 99, j = 179, j^{-1} \bmod p-1 = 151$，计算：

$$\begin{aligned}\gamma &= -2^{99}132^{179} \bmod 467 = 117\\ \delta &= 117 \times 151 \bmod 466 = 41\\ x &= 99 \times 41 \bmod 466 = 331\end{aligned}$$

那么$(117, 41)$是331的有效签名，可验证：

$$132^{117} \times 117^{41} \equiv 303 \pmod{467}$$

ElGamal签名的已知消息攻击的存在性伪造：

- 已知：p, α, β，Alice对消息x的签名(γ, δ)，满足$\beta^\gamma \gamma^\delta \equiv \alpha^x \pmod p$
- 构造出(x', λ, μ)满足:

$$\beta^\lambda \lambda^\mu \equiv \alpha^{x'} \pmod p \tag{2}$$

把底λ写成三个已知的底α, β, γ的形式：

$$\lambda = \alpha^i \beta^j \gamma^h \bmod p \tag{3}$$

$$\begin{aligned}
\beta^\gamma \gamma^\delta \equiv \alpha^x \pmod p \quad & \Leftrightarrow \quad \gamma = (\beta^{-\gamma}\alpha^x)^{\delta^{-1}} \bmod p\text{，代入(3)} \\
& \Rightarrow \quad \lambda = \alpha^{i+x\delta^{-1}h} \beta^{j-\gamma\delta^{-1}h} \bmod p\text{，代入(2)} \\
& \Rightarrow \quad \beta^\lambda (\alpha^{i+x\delta^{-1}h} \beta^{j-\gamma\delta^{-1}h})^\mu \equiv \alpha^{x'} \pmod p \\
& \Leftrightarrow \quad \beta^{\lambda+(j-\gamma\delta^{-1}h)\mu} \equiv \alpha^{x'-(i+x\delta^{-1}h)\mu} \pmod p
\end{aligned}$$

前面得到

$$\beta^{\lambda+(j-\gamma\delta^{-1}h)\mu} \equiv \alpha^{x'-(i+x\delta^{-1}h)\mu} \pmod p$$

若令两边指数为0，即：

$$\begin{aligned}
&\lambda+(j-\gamma\delta^{-1}h)\mu \equiv 0 \bmod p-1 \\
&\qquad\qquad\Rightarrow\quad \mu=\delta\lambda(h\gamma-j\delta)^{-1} \pmod{p-1} \\
&x'-(i+x\delta^{-1}h)\mu \equiv 0 \bmod p-1 \\
&\qquad\qquad\Rightarrow\quad x'=\lambda(hx+i\delta)(h\gamma-j\delta)^{-1} (\bmod p-1) \\
&\qquad\qquad\qquad \lambda=\alpha^i\beta^j\gamma^h \bmod p
\end{aligned}$$

那么(λ,μ)是x'的有效签名，验证留作习题（习题7.4）

泄露随机数k，将暴露私钥a：
由

$$\delta = (x - a\gamma)k^{-1} \pmod{p-1}$$

得到

$$a\gamma \equiv x - k\delta \pmod{p-1}$$

（1）如果$\gamma = \alpha^k \bmod p$与$p-1$，那么$\gamma$可逆，那么可确定出$a$的值：

$$a = (x - k\delta)\gamma^{-1} \bmod p - 1$$

（2）如果$d = (\gamma, p-1) \neq 1$，也可以把a的范围缩小到d个值上，求解参考下一页。

如果对两条不同的消息x_1, x_2签名时使用了相同的随机数k，Oscar可以计算出k值，从而计算出私钥a，达到完全攻破系统的目的：
假设消息x_1的签名为(γ, δ_1)，消息x_2的签名为(γ, δ_2)，那么我们有：

$$\begin{cases} \delta_1 & = & (x_1 - a\gamma)k^{-1} \bmod p-1 \\ \delta_2 & = & (x_2 - a\gamma)k^{-1} \bmod p-1 \end{cases}$$

$$\Rightarrow x_1 - x_2 \equiv k(\delta_1 - \delta_2) \bmod p-1$$
$$\Leftrightarrow x' \equiv k\delta' \bmod p'$$
$$\Leftrightarrow k \equiv x'\delta'^{-1} \pmod{p'}$$
$$\Rightarrow k = x'\delta'^{-1} \bmod p' + ip', i = 0, 1, \ldots, d-1$$

把k代入$\gamma = \alpha^k \bmod p$找到正确值

其中

$$\begin{cases} d & = & \gcd(\delta_1 - \delta_2, p-1) \\ x' & = & \frac{x_1 - x_2}{d} \\ \delta' & = & \frac{\delta_1 - \delta_2}{d} \\ p' & = & \frac{p-1}{d} \end{cases}$$

相关练习：习题7.1

7.4 ElGamal 签名方案的变形

- ElGamal签名方案的缺点：
  - 安全方面：存在性伪造
  - 为了无法计算出离散对数问题，需要模p长度至少需要1024比特，这样导致签名达到2024比特，不利于应用到存储有限的智能卡中。
- 1989年Schnorr提出的ElGamal签名方案的变形可大大缩短签名长度。
- 数字签名算法(DSA)吸收了Schnorr签名方案的一些设计思想，是ElGamal签名的另一种变形。它发表于1994年5月，并于1994年12月1日采纳为标准。
- ECDSA是DSA在椭圆曲线上的应用变形。

Schnorr签名方案对ElGamal签名方案的两点改进：

- 为了缩短签名长度，所基于的离散对数问题是在$(\mathbb{Z}_p^*, \cdot)$上的q阶子群的，而不是群$(\mathbb{Z}_p^*, \cdot)$上。
  - 取α的阶为q，而不是$p-1$(即本原元)。
  - 根据群论的相关结论，需$q|p-1$
  - $\alpha^u \equiv \alpha^v \pmod{p} \Leftrightarrow u \equiv v \pmod{q}$(由课件引理2)，这样可以缩短指数部分存储长度（即$u, v < q$而不是$< p-1$）
  - 安全性基于假设：在特定的Z_p^*子群上求解离散对数是困难的。

- 为了避免存在性伪造，作如下改进：
  - ElGamal的签名验证中判定以下等式是否成立：

$$\beta^{\gamma}\gamma^{\delta} \equiv \alpha^{x} \pmod{p}$$

改为判定以下等式是否成立：

$$\alpha^{\delta-k} \equiv \beta^{\gamma} \pmod{p} \Leftrightarrow \delta - k \equiv a\gamma \pmod{q} \tag{4}$$

伪造ElGamal签名的方法是令判定等式 (4)的两边指数mod$q$为0，即：

$$\begin{cases} \delta - k \equiv 0 \pmod{q} \\ \gamma \equiv 0 \pmod{q} \end{cases} \tag{5}$$

（续）为了抵抗通过解上面方程 (5)来伪造签名，即构造出(x, γ, δ)满足方程 (5)，在指数中引入关于x的*Hash*函数：

$$\gamma = h(x||\alpha^k \bmod p)$$

签名中的另外一个参数δ通过求解判定等式(4)得到：

$$\delta = k + a\gamma \bmod q$$

这样我们构造好Schnorr签名方案的签名部分：x的签名(γ, δ)为

$$\begin{cases} \text{选择随机数} k, 1 \leq k \leq q-1 \\ \gamma = h(x||\alpha^k \bmod p) \\ \delta = k + a\gamma \bmod q \end{cases}$$

（续）现在我们看到判定等式(4)中包含了只有签名者秘密选择的随机数k，因此不能直接作为签名验证的判定式子，需要改进判定等式。对等式(4)移位得到：

$$\alpha^k \equiv \alpha^\delta \beta^{-\gamma} \pmod p$$

把上面的等式代入签名中γ的计算公式得到：

$$\gamma = h(x \| \alpha^\delta \beta^{-\gamma} \bmod p)$$

该等式的所有参数都是公开参数，可作为签名验证的判定等式。

通过上面分析，我们获得了Schnorr签名方案

签名体制7.3 Schnorr签名方案

- 公钥(p, q, α, β)：p为素数，q为素数且$q|p-1$。$\alpha \in \mathbb{Z}_p^*$的阶为$q$，$\beta = \alpha^a \pmod p$。
- 私钥a：$0 \leq a \leq q-1$
- 签名：对消息$x \in \{0,1\}^*$（可为任意长度的比特串）的签名(γ, δ)如下计算

$$\begin{cases} \text{选择随机数}k, 1 \leq k \leq q-1 \\ \gamma = h(x||\alpha^k \bmod p) \\ \delta = k + a\gamma \bmod q \end{cases}$$

- 验证：对消息$x \in \{0,1\}^*$和$\gamma, \delta \in \mathbb{Z}_q$，如下验证：

$$ver_K(x, (\gamma, \delta)) = true \Leftrightarrow h(x||\alpha^\delta \beta^{-\gamma} \bmod p) = \gamma$$

q阶元素α的构造：$\mathbb{Z}_p^*$的本原元可根据数论性质构造得到。如果ρ是$\mathbb{Z}_p^*$的本原元，那么取$\alpha = \rho^{\frac{p-1}{q}}$是$q$阶元素。

根据上面对Schnorr签名方案的构造思路的分析，我们可以看到，Schnorr签名构造的出发点是判定式子 (4)，如果对式 (4)的指数参数重新组合，或者修改正负，可以构造出更多的类似于Schnorr签名的方案，如修改为

$$\alpha^{\delta} \equiv \beta^{k-\gamma} \pmod{p} \Leftrightarrow \delta \equiv a(k-\gamma) \pmod{q} \tag{6}$$

Example

例7.3 假定ElGamal签名的公钥私钥为：

- 私钥$a = 75$
- 公钥(p, q, α, β)：
  取$q = 101$，$p = 78q + 1 = 7879$，3是Z^*_{7879}的本原元，因此取

$$\alpha = 3^{78} \bmod 7879 = 170, \beta = \alpha^a \bmod 7879 = 4567$$

- 如果Alice要对消息x签名，她可如下计算：

1.选择随机数$k = 50$
2.计算$\alpha^k \bmod p = 170^{50} \bmod 7879 = 2518$
3.计算$\gamma = h(x||(2518)_2)$, 其中$(2518)_2$表示2518的2进制表示。为了方便解释，假定$\gamma = h(x||(2518)_2) = 96$
5.$\delta = k + a\gamma \bmod q = 50 + 75 \times 96 \bmod 101 = 79$

因此，签名为$(96, 79)$

- 验证过程，计算：

$$\alpha^{-\delta}\beta^{\gamma} \bmod p = 170^{79} \times 4567^{-96} \bmod 7879 = 2518$$

然后检查$h(x||(2518)_2) = 96$，因此通过验证。

数字签名算法（DSA）是ElGamal签名方案的改进。改进点：

- 和Schnorr签名方案一样，为了缩短签名大小，不是在群$(\mathbb{Z}_p^*, \cdot)$上计算，而是在$(\mathbb{Z}_p^*, \cdot)$的q阶子群上计算，即把本原元α改为q阶元素。
- ElGamal直接对消息x签名，DSA对SHA-1(x)签名。

	ElGamal签名方案	DSA
公钥:	p, α, β	p, q, α, β
私钥:	$a \in [0, p-1]$	$a \in [0, q-1]$
限制条件:	$\beta = \alpha^a \bmod p$	$\beta = \alpha^a \bmod q$
α要求：	为mod$p$本原元， 即阶为$p-1$	q阶元素 且$q \mid p-1$
$\gamma =$	$\alpha^k \bmod p$	$(\alpha^k \bmod p) \bmod q$
$\delta =$	$(x - a\gamma)k^{-1} \bmod p-1$	$(\text{SHA-1}(x) + a\gamma)k^{-1} \bmod q$
验证等式：	$\beta^\gamma \gamma^\delta \equiv \alpha^x \bmod p$	$e_1 = \text{SHA-1}(x)\delta^{-1} \bmod q$ $e_2 = \gamma\delta^{-1} \bmod q$ $(\alpha^{e_1}\beta^{e_2} \bmod p) \bmod q = \gamma$

密码体制7.4 数字签名算法(DSA)

- 公钥(p, q, α, β)：p为L比特长素数，且$L \equiv 0 \pmod{64}, 512 \leq L \leq 1024$，$q$为160比特的素数且$q | p-1$。$\alpha \in \mathbb{Z}_p^*$的阶为$q$，$\beta = \alpha^a \pmod{p}$。
- 私钥a：$0 \leq a \leq q-1$
- 签名：对消息$x \in \{0,1\}^*$（可为任意长度的比特串）的签名(γ, δ)如下计算：

$$\begin{cases} \text{选择随机数} k, 1 \leq k \leq q-1 \\ \gamma = (\alpha^k \bmod p) \bmod q \\ \delta = (\text{SHA-1}(x) + a\gamma)k^{-1} \bmod q \end{cases}$$

- 对于消息x及签名$\gamma, \delta \in \mathbb{Z}_q^*$，验证如下进行：

$$\begin{cases} \text{计算} & e_1 = \text{SHA-1}(x)\delta^{-1} \bmod q \\ & e_2 = \gamma\delta^{-1} \bmod q \\ ver_K(x(\gamma, \delta)) = true & \Leftrightarrow (\alpha^{e_1}\beta^{e_2} \bmod p) \bmod q = \gamma \end{cases}$$

DSA的验证算法的有效性留给同学们作为练习，即证明：如果$\gamma, \delta \in \mathbb{Z}_q^*$是根据DSA签名算法计算得到的$x$ 的签名，那么能够通过DSA的验证算法。

DSA的优缺点：

- 优点：签名长度短，模p长度一样的情况下，DSA的签名长度比ElGamal和Schnorr都短。

	ElGamal	Schnorr	DSA
签名长度	$2\log_2 p$	$\log_2 p + \log_2 q$	$2\log_2 q$

- 缺点：
  - 美国NIST对DSA的评选过程不公开，受到公众的不信任。
  - 最初的模p长度定为512比特，许多人认为过短不够安全，为此NIST对此做了修改，允许模p长度可变。

其它功能的签名方案(不做要求)

前面所介绍的ElGamal、Schnorr和DSA签名方案为基本的签名方案，简要的说，是根据签名算法的基本要求而构造的：

- 不可伪造性：任何人都无法伪造Alice的签名
- 不可否认性：Alice无法否认她的签名
- 可验证性：任何人都可以验证Alice的签名

其它功能的签名方案

但基本的签名算法还不能够满足不同的安全需要，如：

- 不可否认签名： 某些情况下Alice并不希望她的签署的文档到处复制和分发，如签名内容涉及一些敏感信息。这样Alice希望她配合才能够验证签名，否则别人无从判断是否为Alice签署的文档。我们将详细介绍（见7.6节）
- 代理签名：Alice为一个公司的总裁，工作忙碌需要助手代理签名一些文件。
- 盲签名： Alice给Bob支票来支付购买汽车，为此她让银行开支票，支票需要银行的签字认可，但Alice又不希望银行了解支票的金额。
- 多重签名：Alice和Bob签署的合同需要两人的签名。
- 安全增强型签名，如：fail-stop签名、前向安全签名。。。
- 可证明安全签名，即在某些假设下，可证明是计算安全的，如：Lamport签名、全域Hash
- 还有更多应用场合。如果你能够提出一个没有人提过的签名应用场合，将会是一个创新工作。

- 不可否认签名由Chaum和Antwerpen在1989年提出。
- 不可否认签名的应用场合：某些情况下Alice并不希望她的签署的文档到处复制和分发，如签名内容涉及一些敏感信息。这样Alice希望她配合才能够验证签名，否则别人无从判断是否为Alice签署的文档。
- 不可否认签名由三部分组成：
  - 签名算法
  - 验证协议：用于证明是Alice的有效签名，需要Alice和验证者交互完成。
  - 否认协议：用于证明不是Alice的签名，需要Alice和验证者交互完成。如果Alice不愿意完成否认协议，则认为Alice 默认了签名是有效的。如果Alice在否认协议中故意给出错误的数据，她欺骗的概率将是可忽略的。

- 由此可见不可否认签名和普通的签名方案的区别为：
  - 普通签名方案证明和否认签名有效性是由验证协议一起完成的，而不可否认签名分别由验证协议和否认协议完成。
  - 普通签名方案的验证算法不是交互的，不可否认签名的验证和否认协议是交互的。

密码体制7.8 Chaum-van Antwerpen签名方案

- 公钥(p,q,α,β)：$p=2q+1, q$都为素数，α是$\mathbb{Z}_p^*$的q阶元素，$\beta=\alpha^a \bmod p$
- 私钥a：$1\le a\le q-1$
- 令G表示$\mathbb{Z}_p^*$的q阶子群，即$G=\{0,\alpha,\alpha^2,\ldots,\alpha^{q-1}\}$。
- 签名：对消息$x\in G$，签名为

$$y=sig_K(x)=x^a \bmod p$$

- 验证协议：Alice如下向Bob证明y是x的有效签名：
  1. Bob随机选择$e_1,e_2\in\mathbb{Z}_q$
  2. Bob计算$c=y^{e_1}\beta^{e_2} \bmod p$，并将它发送给Alice
  3. Alice计算$d=c^{a^{-1} \bmod q} \bmod p$并将它发送给Bob
  4. Bob接受y是x的合法签名当且仅当$d\equiv x^{e_1}\alpha^{e_2} \pmod p$时

密码体制7.8 Chaum-van Antwerpen签名方案

- 否认协议：Alice如下向Bob证明y不是x的有效签名：
  1. Bob随机选择$e_1, e_2 \in \mathbb{Z}_q^*$
  2. Bob计算$c = y^{e_1}\beta^{e_2} \bmod p$并将它发送给Alice
  3. Alice计算$d = c^{a^{-1} \bmod q} \bmod p$并将它发送给Bob
  4. Bob验证$d \not\equiv x^{e_1}\alpha^{e_2} \pmod{p}$
  5. Bob随机选择$f_1, f_2 \in \mathbb{Z}_q^*$
  6. Bob计算$C = y^{f_1}\beta^{f_2} \bmod p$并将它发送给Alice
  7. Alice计算$D = c^{a^{-1} \bmod q} \bmod p$并将它发送给Bob
  8. Bob验证$D \not\equiv x^{f_1}\alpha^{f_2} \pmod{p}$
  9. 当且仅当$(d\alpha^{-e_2})^{f_1} \equiv (D\alpha^{-f_2})^{e_1} \pmod{p}$，Bob推断签名$y$是伪造的

说明：

- 步骤1-4相当于一次验证过程，步骤5-8也是一次验证过程
- 如果步骤1-4和5-8的验证都不通过，那么转入步骤9验证Alice是否作弊。如果Alice作弊（故意给出错误的数据)而导致两次验证过程都不通过，那么她的作弊行为将会在步骤9发现。