- 1.不可否认签名
- (1)产生系统参数
- 设p是一个大素数，并且p=2q+1，其中q也是素数，Z_p^*中的离散对数是难解的
- 随机取Z_p^*中的阶为q的元素g，由此构造Z_p^*的q阶乘法子群G(即g为G的生成元)，
- 在{1,2…,q-1}中随机选择作为SK，
- 令$PK=g^{SK} \bmod p$，
- 将PK与p,g一起作为公钥公开，而SK作为签名用户的秘密密钥。(离散对数难解问题)

- (2)签名过程
- 签名用户要对消息$m \in G$签名，可用其秘密密钥计算$s=(m)^{SK} \bmod p$，把s作为对m的签名连同m一起发给对方。但这个s事实上是无法被其他人证实的。
- (3)验证协议：
  - 接收方收到s和m后，选择随机数$k_1,k_2 \in Z_p^*$，并计算$c=s^{k1}PK^{k2} \bmod p$，把c送给签名者。
  - 签名者在收到c后计算$d=c^{SK^{-1} \bmod q} \bmod p$，并把d送给签名的接收者。

接收方计算 $t=m^{k_1}g^{k_2} \bmod p$，当且仅当 $t=d$ 时可确定 s 是一个对 m 的合法签名。

- 定理4.1：如果$s \neq (m)^{SK} \bmod p$，则接受者以概率1/q接收s作为m的签名。
- (4)否认协议
- 该协议的作用是签名方要使接收方确信所接收到的签名信息是其他人伪造的。也就是说当签名方声明他没有对信息m签名，则必须执行该协议，并通过协议向接收方证明签名是伪造的。若签名方仅声明没有对信息m签名，又拒绝执行该协议，则就认定签名方对信息m签名了。该协议由两轮验证协议组成。

①接收方选择随机数 $k_1,k_2 \in Z_p^*$，并计算 $c=s^{k1}PK^{k2} \bmod p$，把 c 送给签名者。

② 签名者在收到 c 后计算 $d = c^{SK^{-1} \bmod q} \bmod p$，并把 d 送给签名的接收者。

③ 接收方计算 $t=m^{k_1} g^{k_2} \bmod p$，验证 t 是否不等于 d。

④ 接收方选择随机数 $f_1,f_2 \in Z_p^*$，并计算 $C=s^{f1}PK^{f2} \bmod p$，把 C 送给签名者。

⑤ 签名者在收到 C 后计算 $D = C^{SK^{-1} \bmod q} \bmod p$，并把 D 送给签名的接收者。

⑥ 接收方计算 $T=m^{f_1} g^{f_2} \bmod p$，验证 T 是否不等于 D。

⑦ 接收者宣布 s 是伪造的当且仅当 $(dg^{-k_2})^{f_1} \equiv (Dg^{-f_2})^{k_1} \bmod p$。

定理 4.2：(1)如果 s≠$(m)^{SK}$ mod p，且接收方和签名方都遵守否认协议，则$(dg^{-k_2})^{f_1} \equiv (Dg^{-f_2})^{k_1} \bmod p$。

(2)假设 s=$(m)^{SK}$mod p，且接收方遵守否认协议，如果d≠$m^{k_1}g^{k_2} \bmod p$，D≠$m^{f_1}g^{f_2} \bmod p$，则$(dg^{-k_2})^{f_1} \neq (Dg^{-f_2})^{k_1} \bmod p$的概率是 1-$\frac{1}{q}$。(该结论说明相等概率是 1/q,即明明是签名人的签名，却被成功否认的概率为 1/q)。

- 2.盲签名
- 在通常的数字签名中，总是要知道文件内容然后对该相关信息进行签名，但有时需要某人对一个文件签名，但又不能让他知道文件内容
- 如电子商务中，用户只要银行对资金划转签署同意，而不能知道划拨多少钱，干什么，支付给什么人，因为这是个人的秘密。
- 称这类签名为盲签名。其基本协议是一个所谓的完全盲签名。
- 完全盲签名的具体方案是：A要B对文件签名，但不想让他知道所签的是什么，而B并不关心其签署的内容，只是要确保在需要时可以仲裁，

- 该协议如下：
- (1)A取一文件并乘上随机因子，得到的结果称为盲文件，而随机值则称为盲因子。
- (2)A将盲文件送给B。
- (3)B对盲文件用签名算法签名并送给A。
- (4)A用盲因子除签名结果可得B对原文件的签名。
- 要说明的是，(4)要成立，必须保证签名算法和乘法函数是可换的，否则上述的(1)和(4)必须适当修改，以确保A能得到B对原文件的签名。

- 完全盲签名应具有如下的特点：
- B对文件的签名合法，即能够证明B确实对该文件签了名；
- B不能将所签的文件与实际上签的文件联系起来，即使他保存所有签过的文件，也不能得到文件的真实内容，当然攻击者所得信息更少。
- 但由于B不知道自己所签名的内容，就有可能使A让B签署任何自己所想要的文件，甚至可以让B对诸如“B欠A100万元”等，因不能保证A永远是诚实的..
- 采用分割-选择技术，可以使B知道相信他所签的内容是合理的，但又不知道所签的内容。

- 分割-选择技术的盲签名协议：
- (1)每个成员准备n份文件，各用不同的掩护名字，以得到外交豁免权。
- (2)成员以不同的盲因子盲化每个文件，得到n个盲文件，并把它们送给谍报机构。
- (3)谍报机构从n个盲文件中随机选取n-1，询问成员它们每个对应盲因子。
- (4)成员送出所对应的盲因子给谍报机构。
- (5)谍报机构从n-1盲文件中移取盲因子，得到原文件并检查其正确性。
- (6)谍报机构对n个盲文件中的另一个盲文件签名并送给该成员。
- (7)成员移取盲因子，即得其隐蔽身份外交豁免权的签名文件。

- 该协议能有效防止成员的欺诈行为。
- 因为成员要欺骗成功，必须知道哪个文件不被检验，其成功概率是1/n。
- 当然其成员也可以这样做，对每个文件都准备两个不同内容的文本对，并找到两个盲因子，使得它们的盲文本是相同的。
- 这样若不要求看原文件并签名，就可用对应蓄意制造的文件的盲因子得到签名；若要求检验原文件，就可用对应合法文件的盲因子传送给谍报机构。
- 当然这在理论上是可能的，而在实际上通常是在计算上是难的。

- 完全盲签名方案——基于DSA变形的盲签名方案：
- 基于DSA变形的签名方案
- (1)密钥的产生：随机选取一个大素数p(其中512≤||p||≤1024,且按64bit递增)，取字长为160bit的p-1素因子q，在{2,3,…,p-2}中选取a，使得a^q =1mod p。
- 用户A在{1,2,…,q-1}内任取元素SK作为秘密密钥，并计算PK=a^{SK}mod p, 将PK作为其公开钥，与p,q和a一起公开。

- (2)签名过程
- 用户A要对消息m(m<p)签名，可用其秘密密钥进行运算，并增加一随机数，其过程如下
- i)在0与q之间选择秘密随机数k
- ii)计算$r=(a^k \bmod p) \bmod q$
- $s=(km+r*SK) \bmod q$
- iii)取(r,s)作为m的签名连同m一起发给对方。
- (3)验证过程：
- 接收方收到(r,s)和m后，计算$v=((a^s PK^{-r})^{m^{-1}} \bmod p) \bmod q$，然后判断v是否与r相同，由此即可确定(r,s)是否用户A对m的签字。

- 基于上面DSA变形的盲签名方案:
- (1)密钥的产生：随机选取一个大素数p(其中512≤||p||≤1024,且按64bit递增)，取字长为160bit的p-1素因子q，在{2,3,…,p-2}中选取a，使得a^q =1mod p。用户A可在{1,2,…,q-1}内任取元素SK作为秘密密钥，并计算PK=a^{SK}mod p，将PK作为其公开钥，与p,q和a一起公开。

- (2)签名过程
- 当用户B要用户A对消息m(m<p)盲签名时，双方执行以下协议：
- A:i)在1与q之间(即在{1,2,…,q-1})选择秘密随机数k
- ii)计算K=(a^k mod p),并验证K与q是否互质，若是则将K送给B，否则重新执行i)
- B:i)验证K与q是否互质，若不是则要A重新送出K
- ii) 在1与q之间选择秘密随机数α,β
- iii)计算R=$K^{\alpha}a^{\beta}$mod p,并验证R与q是否互质，若不是则重新执行ii)
- iv)计算m'=αmKR^{-1}modq，将m'送给A

- A:计算s'=(km'+K*SK)modq，将s'送给B
- B:i)计算$s=(s'*R*K^{-1}+\beta*m)\bmod q$,
- $r=R \bmod q$
- ii)取(r,s)作为A对m的签名。
- (3)验证过程：
- 验证方进行以下计算：
- i)计算$v=((a^{s}PK^{-r})^{m^{-1}} \bmod p) \bmod q$，然后判断v是否与r相同，由此即可确定(r,s)是否用户A对m的签字。

- 3.代理签名
- 1996年，Mambo、Usuda和Okamoto首先提出了代理签名的概念，所谓代理签名就是指：在一个代理签名方案中，一个被指定的代理签名者可以代表原始签名者产生有效的签名。代理签名在电子商务、电子政务、电子现金、电子投票和移动代理等方面有广泛的应用前景。

- (1)某个公司的经理在某一段时间内，由于某些原因不能行使数字签名权，委托他的秘书代表他在这段时间内行使他的数字签名权。
- (2)某一软件公司为了向客户证实它出品的程序的可靠性，需要以公司的名义对所有这些程序进行数字签名。由于程序太多，经理无法亲自检测每个程序，一个可行的方法是，公司经理将代表公司生成数字签名的权利委托给每个程序员，让他们以各自的名义为他们创作的程序生成数字签名。
- (3)某一总行授予所属支行签发电子货币的权力，让他们以总行的名义签发电子货币，但又不让他们获得总行的签发密钥，以免其滥发行电子货币。

- 代理签名的定义：设A和B分别是某个数字签名体(SK，PK，M，S，GenKey，Sign，Ver)的原始签名者和代理签名者，其中，SK是私钥集合，PK是公钥集合，M是消息集合，S是签名集合，GenKey表示产生签名密钥的算法的集合，Sign是签名算法集合，Ver是签名验证算法的集合，他们的私有密钥和公开密钥对分别是(x_A,y_A)，$(x_B,y_B)\in SK\times PK$。如果满足以下条件：
- (1)A利用自己的秘密密钥x_A计算出一个数σ，并且将σ秘密地交给B。
- (2)包括B在内的任何人在求解x_A时，σ都不能有任何的帮助。
- (3)代理签名人B利用σ和自己的私有密钥x_B产生新的签名密钥$\sigma_{A\to B}\in SK$，利用签名算法产生代理签名$Sign(\sigma_{A\to B}，m)$。

- (4) 存在一个公开的验证算法 $Ver_{A\to B}$：$PK\times S\times M\to\{True，False)$，使得对于任何$s\in S$和$m\in M$，都有：
- $Ver_{A\to B}(y_A,s,m)=True\Leftrightarrow s=Sign(\sigma_{A\to B},m)$。
- (5)任何人在试图求出x_A、x_B、σ或$\sigma_{A\to B}$时，任何数字签名$Sign(\sigma_{A\to B}，m)$都不会对求解有任何帮助。
- 那么我们就称用户A将他的数字签名权力委托给了用户B，并且称A为原始签名者，B为A的代理签名者，称$\sigma_{A\to B}$为代理签名密钥，称以$\sigma_{A\to B}$为签名密钥对消息$m\in M$生成的数字签名$Sign(\sigma_{A\to B}，m)$为A的代理签名。

- 代理签名方案应该满足下面的基本性质：
- (1)签名的可验证性。验证者能够相信原始签名者已经同意代理签名者可以代表他自己行使签名权。
- (2)签名的不可伪造性。除了原始签名者和指定的代理签名者能够代表原始签名者产生有效的代理签名,其他人无法产生签名。
- (3)签名的不可否认性。一旦代理签名者代,表原始签名者对消息产生了有效的代理签名，就不能向原始签名者否认他所签的有效的代理签名。
- (4)签名的可区分性。任何人都可以区分代理签名和正常的原始签名者的签名。
- (5)代理签名者身份的确认性。原始签名者能够从代理签名中确认代理签名者的身份。
- (6)代理密钥的依赖性。代理密钥必须是由原始签名者的秘密密钥产生，即代理密钥必须依赖原始签名者的秘密密钥。
- (7)可注销性。如果原始签名人希望代理签名人只能在一定时间区间内拥有生成代理签名的能力，那么必须能够让代理签名人的代理签名密钥在指定的时刻失去作用。

- 为了体现对原始签名者和代理签名者的公平性，Lee等对其中的一些性质给出了更强的定义：
- (1)强不可伪造性。只有指定的代理签名者能够产生有效的代理签名，原始签名者和没有被指定为代理签名者的任何人都不能产生有效的代理签名。
- (2)强不可否认性。一旦代理签名者产生了有效的代理签名后，他就不能向任何人否认他所签过的代理签名。
- (3)强可鉴别性。通过代理签名，任何人都能够确定代理签名人的身份。
- (4)防止滥用。应该确保代理密钥不能被用于其他目的。为了防止滥用，代理签名者的责任应当被具体确定。

- 代理签名可分为三大类：完全代理签名、部分代理签名和具有证书的代理签名。
- (1)完全代理签名
- 在完全代理签名中，原始签名者直接把自己的签名密钥通过安全信道发送给代理签名者，他们能产生相同的签名。
- 签名不可区分的，不能阻止签名滥用。也不具有可识别性和不可否认性。不适用于商业应用。
- (2)部分代理签名
- 在部分代理签名中，原始签名人使用自己的私钥X_A，产生一个签名密钥σ，并把σ通过安全信道传送给代理签名人，这时，代理签名人的代理签名和原始签名人的签名是可以区分的，

- (3)具有证书的代理签名
- 具有证书的代理签名有授权代理签名和持票代理签名两类。
- 授权代理签名：在授权代理签名中，原始签名者用签名密钥使用普通的签名方案签一个文件，然后把产生的证书发给代理签名者。
- 持票代理签名：证书由消息部分和原始签名者对新产生的公钥的签名组成。原始签名者把新产生的公钥所对应的密钥以安全的方式发送给代理签名者。
- 目前，主要集中在部分代理签名和具有证书的代理签名，并且两者往往是交叉的，
- 具有证书的部分代理签名具备了部分代理签名和具有证书的代理签名的双方的优点，
- 还有根据代理签名方案所基于的数学难题，可以分为基于离散对数问题的代理签名方案和基于大数分解问题的代理签名方案。
- 根据代理签名所具有的特殊性质，可以分为代理多重数字签名方案、前向安全代理签名方案、匿名代理签名方案和定向代理签名方案等等。

- M．U-O代理签名方案
- 1996年，Mambo等提出了代理签名的概念，并给出了实现代理签名的实际例子，具体过程为：
- 1．初始化过程
- (1)p是一个大素数，q是(p-1)的一个大素数因子。
- (2)g是Z^*_p的一个q阶元。
- (3)原始签名者O选择随机数$x_O \in Z^*_q$，x_O作为私钥秘密保存，计算$y_O=g^{x_O} \bmod p$，y_O作为公钥公开。

- 2．代理授权过程
- (1)O随机选取$k \in Z^*_q$，并计算$r=g^k \bmod p$
- $s_O=x_O+kr \bmod p$
- (2)O通过秘密信道把(s_O，r)传送给代理签名者P。
- (3)P验证等式$g^{so}=y_O r^r \bmod p$，如果等式成立，则取$x_{OP}=s_O$为代理签名密钥，如果等式不成立，则拒绝接受，协议终止或让O重新发送。

- 3．代理签名过程
- 当代理签名者P代理原始签名者O在文件m上签名时，P利用普通的数字签名算法Sign，使用代理签名密钥x_{OP}代替原始签名密钥x_O对m进行签名$s=Sign_{x_{OP}}(m)$，有效的代理签名是(m，s，r)。

- 4．签名验证过程
- 验证者首先计算代理公钥$y_{OP}=y_O r^r \bmod p$，然后利用与签名算法Sign相对应的普通签名验证算法Ver，使用代理公钥y_{OP}代替原始签名者的公钥y_O验证代理签名的有效性。即如果$Ver(y_{OP}, s, m)=True$，则代理签名为原始签名者有效的代理签名，否则为无效的签名。
- 缺陷：原始签名者授权签名中没有包含关于代理签名者身份或者证书的有关信息；
- 代理签名者可以把授权签名交给第三者，第三者也能产生一个正确的代理签名；
- 代理签名是否由代理签名者所做，无法证实。

- **Lee代理签名方案**
- **在代理签名中添加授权书，Kim等提出了一种新形式的代理签名方案。该方案中因为具有原始签名者的授权书，因而可以对代理权限的有效期、代理授权的使用范围等信息进行规定，可以防止代理签名的滥用，但原始签名者和代理签名者身份是一样的，因而通过代理签名无法知道是原始签名者还是代理签名者作了签名。Lee在Kim等方案的基础上进行了改进，具体的过程：**

- 1．初始化过程
- (1)p是一个大素数，q是(p-1)的一个大素数因子。
- (2)g是Z^*_p的一个q阶元。
- (3)h()是安全的单向散列函数。
- (4)m_w是代理授权书，是一个包括原始签名者标识、代理签名者标识、代理权限的有效期和代理签名信息的范围等信息的文件。.
- (5) 原始签名者O选择随机数$x_O \in Z^*_p$，x_O作为私钥秘密保存，计算$y_O=g^{x_O} \bmod p$，y_O作为公钥公开。
- (6)代理签名者P选择随机数$x_P \in Z^*_q$，x_P作为代理签名者的私钥秘密保存，计算$y_P=g^{x_P} \bmod p$，y_P作为其公钥公开。

- 2．代理授权过程
- (1)O随机选取$k \in Z^*_q$，并计算$r=g^k \bmod p$
- $s_O=x_O h(m_w, r)+k \bmod p$
- (2)O通过秘密信道把(m_w，s_O，r)传送给代理签名者p。
- (3)P验证等式$g^{s_O}=y_O^{h(m_w, r)} r \bmod p$，如果等式不成立，则拒绝接受，协议终止或让O重新发送，若等式成立，则进行下一步。
- (4)P计算$x_{OP}= s_O+x_P \bmod q$并把x_{OP}作为代理签名私钥。

- 3．代理签名过程
- 如果文件m符合代理授权书m_w的要求，代理签名者P利用普通的签名算法Sign，使用代理签名密钥x_{OP}代替原始签名私钥x_O对m进行签名$s=Sign_{x_{OP}}(m)$，
- 有效的代理签名为(m，s，r)。

- 4．代理签名验证过程
- 第一步，验证者首先计算代理公钥$y_{OP}=y_O^{h(m_w, r)} r^{y_P} \bmod p$，然后利用与签名算法Sign相对应的签名验证算法Ver，使用代理公钥y_{OP}代替原始签名者的公钥y_O验证代理签名的有效性。即$Ver(y_{OP}, s, m)=True$是否成立。
- 第二步，验证者检查原始签名者和代理签名者的标识、文件m、代理有效期是否符合代理授权书m_w的要求。
- 如以上两步验证都能通过，则认为这个代理签名是有效的，否则，上述两步中有任何一个验证通不过，则这个代理签名是无效的。

- 实际使用时的签名模式
- 由于各种签名体制都是基于公钥密码体制的，其计算效率不高，
- 对于信息量较大的信息进行签名时，实际使用时通常是对信息m先用Hash函数进行压缩，然后对压缩信息进行签名。
- 而验证方采用同样的Hash函数对m进行压缩，然后采用验证算法进行验证。

4.2 Hash函数

- 散列函数的选取不能减弱签名方案的安全性，它必须能够防止他人伪造。因此一个好的散列函数应该具有以下特性：
- 弱无碰撞：给定一个消息x，找到一个满足$h(x') = h(x)$ 的消息$x' \neq x$ 是计算上不可行的；
- 强无碰撞：找到满足$h(x') = h(x)$ 并且$x' \neq x$ 的消息x'和 x是计算上不可行的；
- 单向：给定一个消息摘要z，找到一个满足$h(x) = z$ 的消息x 是计算上不可行的。

- 4. 2. 1生日攻击
- 生日攻击是一种比较有效的攻击Hash函数的方法。为了了解生日攻击，先看下面几个问题。
- 例1：在一个会场中，要求找到一个人使得其生日与某人生日相同的概率超过0.5，问会场中至少需要多少人？
- 设需要t个人
- $t \geq 183$

- 例2：在一个会场中，要求至少有两个人生日相同的概率超过0.5，问会场中至少需要多少人？
- 解：设需要t个人。
- $t \geq 23$。
- 不同的提法其结果相差很大。
- 就采用类似第二个例子的方法来攻击Hash函数，
- 该攻击方法不涉及Hash函数的结构，可攻击任何Hash函数。

- 生日攻击的方法：
- 设h是A到B的Hash函数，且$|A| \geq 2|B| = 2n$。
- 显然至少有n个碰撞(n个值，n个碰撞)，攻击者关键是如何找到这些碰撞。
- 生日攻击的方法就是随机选择k个不同的元素$x_1, x_2, \ldots, x_k \in A$，计算$y_i = h(x_i)(1 \leq i \leq k)$，然后确定其中是否有值相同的(即所谓的碰撞)，现在就是考察需要多少个k能以概率ε找到一个碰撞。

- 由于对于函数h(x)，若$y \in B$的源象集基不是近似相等，则找到一个碰撞的概率将增大，因此散列函数的每个象的源象集基数应该是近似相等的。又因为$x_i(1 \le i \le k)$是随机选的，因此可把$y_i=h(x_i)(1 \le i \le k)$看作为B中的随机元素。
- y_1任意选择，$y_2 \neq y_1$的概率为1-1/n，$y_3 \neq y_1, y_2$的概率为1-2/n，…，$y_k \neq y_1, y_2 \ldots, y_{k-1}$的概率为1-(k-1)/n，因此没有碰撞的概率是
- $(1-1/n)(1-2/n)\ldots(1-(k-1)/n) \approx e^{-k(k-1)/2n}$
- 至少有一个碰撞的概率是$1-e^{-k(k-1)/2n}$

如果取$\varepsilon=0.5$，则$k \approx 1.17\sqrt{n}$

- 杂凑$n^{1/2}$个A中的随机元素就能以50%的概率找到一个碰撞。对40bit长的信息摘要(即散列函数值为40bit)，则$|B|=2^{40}$，则$k\approx 2^{20}$(约1百万)次随机杂凑就可以以50%的概率找到一个碰撞。
- 因此，为了抵抗生日攻击，散列函数的输出长度建议至少需要128bit，此时生日攻击需要$k\approx 2^{64}$次杂凑，目前使用的散列函数都在128bit及以上。

- Hash函数如何构造，以达到前面的要求？
- 一个最基本必须满足的要求应该是任何输入串中单个比特发生变化，将会导致输出比特串中大约一半的比特发生变化。
- 构造方法主要有以下几种
- 利用某些对称密码体制，设计Hash函数
- 利用某些数学难题假设，设计Hash函数，可以在某些难问题假设下证明是强无碰撞的。
- 直接设计Hash函数。

- 作业:P137 2
- 1.总结代理签名的基本结构和设计思路。