密码学复习

公钥密码体系

RSA加密

数学难题

该算法的数学基础是初等数论中的欧拉定理，其安全性基于大整数因子分解的困难性

密钥的生成

1. 选择两个大素数p和q,（$p \neq q$，需要保密，步骤4以后建议销毁）
2. 计算$n = p \times q, \varphi(n) = (p-1) \times (q-1)$
3. 选择整数 e 使$gcd(\varphi(n)，e) = 1, 1 < e < \varphi(n)$
4. 计算d，使$d = e^{-1} \bmod \varphi(n)$

公钥为{e, n}； 私钥为{d}

加密(用e，n)

明文$M < n$， 密文$C = M^e (\bmod n)$.

解密(用d，n)

密文C， 明文$M = C^d (\bmod n)$

正确性

- **若**$gcd(m, n) = 1$**，**

$C^d \bmod n = (m^e)^d \bmod n = m^{ed} \bmod n \equiv m \bmod n$

- **若**$gcd(m, n) > 1$**，**

书上太麻烦了，不好理解，直接拿CRT中国剩余定理，$ed = 1 \bmod \varphi(n)$,由于$n = p \times q$,

$$C^d \bmod n = (m^e)^d \bmod n = m^{ed} \bmod n = m^{k\varphi(n)+1} \bmod pq$$

$$\begin{cases} m^{k\varphi(n)+1} \bmod p = m(m^{\varphi(p)})^{k\varphi(q)} = m \bmod p \\ m^{k\varphi(n)+1} \bmod q = m(m^{\varphi(q)})^{k\varphi(p)} = m \bmod q \end{cases}$$

所以：

$$C^d \bmod n = m^{k\varphi(n)+1} \bmod pq = m \bmod n$$

总结

密钥生成算法	n：两素数 p 和 q 的乘积（p 和 q 必须保密） $\varphi(n)=(p-1)(q-1)$ 公钥 e：满足 $\gcd(e,\varphi(n))=1$，即 e 与 $\varphi(n)$ 互素
	私钥 d：$d\equiv e^{-1}(\bmod\ \varphi(n))$
加密算法	$c\equiv m^{e}(\bmod\ n)$
解密算法	$m\equiv c^{d}(\bmod\ n)$

对RSA的攻击

- 针对n分解的攻击
  - 试除法
  - 因子分解法（p±1法，连分数法，二次筛法，椭圆曲线筛法，数域筛法）
- 循环攻击
- 同模攻击
- 选择密文攻击
- 低加密指数攻击
- 时间攻击

ElGamal加密

离散对数问题

设p至少是150位的十进制素数，p-1有大素因子。Z_p为有限域，若g为Z_p中的本原元/生成元/原根，有

$$Z_p^* =< g >, \beta \in {Z_p}^* = Z_p \backslash \{0\},$$
$$\text{求唯一的整数} a (0 \le a \le p-2), \text{满足} g^a = \beta (\bmod p)$$
$$\text{记为} a = log_g(\beta)$$

密钥生成

选取大素数p，$g \in Z_p^*$是一个生成元，p, g 作为系统参数所有用户共享

系统中每个用户U都随机挑选整数x，2≤x≤ p - 2，并计算：$y = g^x(modp)$

公钥$\{y, g, p\}$, 私钥$\{x\}$

加密

用户A先把明文M编码为一个在 0 到p - 1之间的整数m； $m \in [0, p-1]$

用户A挑选一个秘密随机数$r(2 \le r \le p{-}2)$,

并计算：

$$c_1{=}g^r(\bmod p)$$
$$c_2{=}my^r(\bmod p)$$

用户A把二元组 (c_1, c_2) 作为密文传送给用户B

解密

用户B接收到密文二元组(c_1, c_2)后，做解密计算：

$$c_2(c_1^x)^{-1} = m \bmod p$$

正确性

$$
\begin{aligned}
c_2 \cdot (c_1^x)^{-1} (\bmod p) &= (my^r)(g^{rx})^{-1} (\bmod p) \\
&= (m * g^{xr})(g^{rx})^{-1} (\bmod p) \\
&= m (\bmod p)
\end{aligned}
$$

总结：

算法特点

- 非确定性：由于密文依赖于加密过程中用户A选择的随机数r，所以加密相同的明文可能会产生不同的密文—概率加密
- 密文空间大于明文空间：明文空间为Z_p^*，而密文空间为$Z_p^* \times Z_p^*$

密钥生成算法	公钥	p：大素数 g：$g<p$ y：$y \equiv g^x (\bmod\ p)$
	私钥	x：$1<x<p-1$
加密算法	r_i：随机选择，$1<r_i<p-1$ 密文：$c_i \equiv g^{r_i} (\bmod\ p)$ $c'_i \equiv m_i y^{r_i} (\bmod\ p)$	
解密算法	明文：$m_i \equiv (c'_i / c_i^x)(\bmod\ p)$	

攻击方法

- 大步-小步法（Giant-step Baby-step）
- 指数积分法（Index Calculus） —更有效，亚指数时间算法

数字签名

概念

数字签名的过程

- 系统初始化过程：产生数字签名方案中的所有系统和用户参数(公开的+秘密的)
- 签名过程：用户利用给定的签名算法对消息签名，签名过程可以公开也可以不公开，但一定包含仅签名者才拥有的秘密信息（签名密钥）
- 验证过程：验证者利用公开的验证方法对给定消息的签名进行验证

RSA签名方案

数学难题

该算法的数学基础是初等数论中的欧拉定理，其安全性基于大整数因子分解的困难性

密钥的生成

1. 选择两个大素数p和q，($p \neq q$，需要保密，步骤4以后建议销毁)
2. 计算$n = p \times q, \varphi(n) = (p-1) \times (q-1)$
3. 选择整数 e 使$gcd(\varphi(n)，e) = 1, 1 < e < \varphi(n)$
4. 计算d，使$d = e^{-1} \bmod \varphi(n)$

公钥为{e, n}； 私钥为{d}

签名算法

设待签名的消息为$m \in Z_n$，利用一个安全的Hash函数h来产生消息摘要$h(m)$,然后签名者 A 用下面算法计算签名$s = h(m)^d (\bmod n)$,则s是消息m的签名。(s,m)发送给B。

验证算法

签名接收者 B 收到m和签名s后,首先,利用上述Hash 函数h计算消息摘要 $h(m)$；然后检验等式$h(m)(\bmod n) = s^e(\bmod n)$是否成立。若成立，则签名有效;否则，签名无效。

正确性验证

因为

$$
\begin{gathered}
s = h(m)^d \bmod n \\
ed = 1 \bmod \varphi(n) \\
\varphi(n) = (p-1)(q-1)
\end{gathered}
$$

所以(详细看上面RSA加密一样)

$$
\begin{aligned}
s^e(\bmod n) = h(m)^{ed} \bmod n = h(m)^{k\varphi(n)+1}(\bmod n) \\
= h(m)(\bmod n)
\end{aligned}
$$

总结

H(M)的重要性

$\boldsymbol{H(M)}$的另一个作用—加快签名速度
对整个消息签名，由于公钥体制速度比较慢，当消息比较长时，签名与验证过程都会相当慢

对消息的Hash值签名，则无论消息多长，签名都只与Hash值的长度有关

攻击方法

- 一般攻击:意义不大
- 利用已有的签名进行攻击:俩个签名组合起来，对策：采用Hash函数
- 利用签名获得明文 对策对数据的Hash值签名

ELGamal签名体系

离散对数问题

设p至少是150位的十进制素数，p-1有大素因子。Z_p为有限域，若g为Z_p中的本原元/生成元/原根，有

$$
\begin{gathered}
Z_p^* =< g >, \beta \in {Z_p}^* = Z_p \backslash \{0\}, \\
\text{求唯一的整数} a (0 \leq a \leq p-2), \text{满足} g^a = \beta (\bmod p) \\
\text{记为} a = log_g(\beta)
\end{gathered}
$$

密钥生成

选取大素数p，$g \in Z_p^*$是一个生成元，p, g 作为系统参数所有用户共享

系统中每个用户U都随机挑选整数x，2≤x≤ p - 2，并计算：$y = g^x (modp)$

公钥$\{y, g, p\}$, 私钥$\{x\}$

签名算法

设待签消息为 m ,签名者选择随机数 $k \in_R Z_p$,计算：

$$r \equiv g^k (\bmod\ p);$$
$$s = [h(m) - xr]k^{-1} (\bmod\ (p-1))$$

则对消息m的数字签名为(r，s),其中h为安全的Hash函数。

验证算法

签名接收者B收到消息 m 和签名(r, s) 后,首先计算 h(m) ,然后验证下列等式是否成立:

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

如等式成立,则签名有效;否则,签名无效.

正确性验证

如果所有算法按步骤执行,则接收者输出签名有效,因为

$$r \equiv g^k (\bmod\ p), s \equiv [h(m) - xr]k^{-1} (\bmod\ (p-1))$$

所以

$$ks \equiv h(m) - xr (\bmod\ (p-1))$$
$$r^s = g^{ks} \equiv g^{h(m)-xr} (\bmod\ p)$$
$$r^s y^r = g^{ks} g^{xr} \equiv g^{h(m)} (\bmod\ p)$$
$$y^r r^s \equiv g^{h(m)} (\bmod\ p)$$

安全性分析

- 非确定性数字签名算法，同一消息M的签名依赖于随机数$\boldsymbol{k}$
- 安全性基于有限域上计算离散对数的困难性
- 随机数$\boldsymbol{k}$不能被泄露(已知$\boldsymbol{k}$可以计算x)
- 随机数$\boldsymbol{k}$不能被重复使用（泄露x ）
- 不使用Hash函数则易受到攻击