

试讲：RSA加密算法

试讲人：刘坤

RSA算法

- ▶ 由MIT的 Rivest, Shamir 和 Adleman 在 1977 提出
- ▶ 是一个分组加密算法
- ▶ 最著名的且被广泛应用的公钥加密体制
- ▶ 理论基础是数论中的论断：要求得到两个大素数的乘积是容易的，但要分解一个合数为两个大素数的乘积在计算上几乎是不可能的。

RSA 密钥产生过程

1. 随机选择两个大素数 p, q
2. 计算 $N=p \times q$
3. 计算小于 N 并且与 N 互质的整数的个数, 即欧拉函数 $\varphi(N)=(p-1)(q-1)$
4. 选择 e 使得 $1 < e < \varphi(N)$, 且 $\gcd(e, \varphi(N))=1$
5. 解下列方程求出 d , $e \times d = 1 \bmod \varphi(N)$ 且 $0 \leq d \leq N$

RSA 密钥产生过程

1. 随机选择两个大素数 p, q
2. 计算 $N=p \times q$
3. 计算小于 N 并且与 N 互质的整数的个数, 即欧拉函数 $\varphi(N)=(p-1)(q-1)$
4. 选择 e 使得 $1 < e < \varphi(N)$, 且 $\gcd(e, \varphi(N))=1$
5. 解下列方程求出 d , $e \times d = 1 \bmod \varphi(N)$ 且 $0 \leq d \leq N$
 - 保密 d , p 和 q (销毁), 公开 N 和 e
 - 公布公钥: $PU=\{e, N\}$
 - 保存私钥: $PR=\{d, N\}$

RSA 密钥生成的计算量

1. 如何得到足够大的随机素数
2. 如何求解方程 $exd=1 \bmod \varphi(N)$

如何得到足够大的随机素数

- 实际应用所采用的方法是：首先，产生一个随机数，然后通过一个概率多项式时间算法检测该随机数是否为素数
- 常用的两个素性测试算法：
 - Solovay-Strassen素性测试
 - Miller-Rabin素性测试

求解方程 $exd=1 \bmod \varphi(N)$

扩展的欧几里得算法(辗转相除法)

例子, 当 $e = 1001$, $\varphi(n) = 3837$ 时方程为
 $x * 1001 = 1 \pmod{3837}$

求解过程:

$$3837 = 3 * 1001 + 834$$

$$1001 = 1 * 834 + 167$$

$$834 = 4 * 167 + 166$$

$$167 = 166 + 1$$

求解方程 $exd=1 \bmod \varphi(N)$

扩展的欧几里得算法(辗转相除法)

例子, 当 $e = 1001$, $\varphi(n) = 3837$ 时方程为
 $x * 1001 = 1 \pmod{3837}$

所以

$$\begin{aligned} 1 &= 167 - 166 \\ &= 167 - (834 - 4 * 167) \\ &= 5 * 167 - 834 \\ &= 5 * (1001 - 834) - 834 \\ &= 5 * 1001 - 6 * 834 \\ &= 5 * 1001 - 6 * (3837 - 3 * 1001) \\ &= 23 * 1001 - 6 * 3837 \end{aligned}$$

RSA使用

公布公钥: $PU=\{e,N\}$ 保存私钥: $PR=\{d,N\}$

- ▶ 加密一个报文 M , 发送方:
 - 获取接收方的公钥 $PU=\{e,N\}$
 - $C=M^e \bmod N$, where $0 \leq M < N$
- ▶ 解密密文 C , 接收方:
 - 用自己的私钥 $PR=\{d,N\}$
 - 计算 $M=C^d \bmod N$
- ▶ 必须满足以下条件:
 - $M^{ed} = M \bmod N$
 - 计算 M^e 和 C^d 是比较容易的
 - 由 e 和 n 确定 d 是不可行的

为什么RSA 可以加解密

▶ 因为 Euler 定理的一个推论:

- ▶ $M^{k\varphi(N)+1} = M \bmod N$

▶ RSA 中:

- ▶ $N=p \cdot q$

- ▶ $\varphi(N)=(p-1)(q-1)$

- ▶ 选择 e & d 使得 $ed=1 \bmod \varphi(N)$

- ▶ 因此 存在 k 使得 $e \cdot d=1+k \cdot \varphi(N)$

▶ 因此

$$C^d = (M^e)^d = M^{1+k \cdot \varphi(N)} = M \bmod N$$

RSA 例子

1. 挑选质数: $p=17$ & $q=11$
2. 计算 $n = pq = 17 \times 11 = 187$
3. 计算 $\varphi(n) = (p-1)(q-1) = 16 \times 10 = 160$
4. 选择 e : $\gcd(e, 160) = 1$; 不妨 $e=7$
5. 求解 d : $d \times e = 1 \pmod{160}$ 且 $d < 160$
 $d=23$ 显然 $23 \times 7 = 161 = 10 \times 160 + 1$

Publish key $PU = \{7, 187\}$

Private key $PR = \{23, 187\}$

RSA 例子

▶ RSA 加密/解密:

▶ $M = 88$ (注意 $88 < 187$)

▶ 加密:

$$C = 88^7 \bmod 187 = 11$$

▶ 解密:

$$M = 11^{23} \bmod 187 = 88$$

模指数运算简化

- ▶ 在RSA密码体制中，加密和解密运算都是模指数运算，即 $C=M^e \bmod N$
- ▶ 可以通过 $e-1$ 次模乘来实现计算，然而，如果 e 越大，其效率会很低下
- ▶ 平方-乘算法可以把计算所需的模乘的次数降低，实现高效算法

求模指数实例子

$$11^{23} \bmod 187 = [(11^1 \bmod 187) * (11^2 \bmod 187) * (11^4 \bmod 187) * (11^8 \bmod 187) * (11^8 \bmod 187)] \bmod 187$$

$$11^1 \bmod 187 = 11$$

$$11^2 \bmod 187 = 121$$

$$11^4 \bmod 187 = 14641 \bmod 187 = 55$$

$$11^8 \bmod 187 = 214358881 \bmod 187 = 33$$

$$11^{23} \bmod 187 = (11 * 121 * 55 * 33 * 33) \bmod 187 = 79720345 \bmod 187 = 88$$

RSA注意

- ▶ RSA加密时，明文以分组的方式加密；每一个分组的比特数应该小于 $\log_2 n$ 比特，即， $M < N$
- ▶ 选取的素数 p 和 q 要足够大，从而乘积 N 足够大，在事先不知道 p 和 q 的情况下，分解 N 是计算上不可行的

结束，谢谢！