密码学

张帆

fanzhang@zju.edu.cn

ZHEJIANG UNIVERSITY

第7章 RSA算法

浙江大學
ZHEJIANG UNIVERSITY

非对称加密

- 非对称加密=公钥加密（public key encryption, PKE）
  - $K_E \neq K_D$
- PKE系统消除了对称加密问题
  - 不需要安全的密钥分发通道=>轻松分配密钥

非对称加密

- 一种PKE方法：
  - 用户保留着私钥K_D
  - 可以将相应的公钥K_E分发给任何想要向他发送加密信息的人
  - 不需要安全通道来发送K_E，甚至可以在一个开放的网站上发布密钥——这是公开的

- 只有私钥K_D才能解密用公钥K_E加密的信息
  - 任何人（K_E是公开的）都可以加密
  - 只有私钥K_D的所有者才能解密

非对称加密要求

- 1.在给定适当密钥的情况下，对消息进行加密或解密必须在计算上很容易
- 2.从公钥导出私钥在计算上一定是不可行的
- 3.从选定的明文攻击中确定私钥在计算上一定是不可行的

密钥对特征

- 1.一个密钥是另一个密钥的互补，它可以撤销另一方提供的加密，例如：

$$D(k_{PRIV}, E(k_{PUB}, P)) = P$$
$$D(k_{PUB}, E(k_{PRIV}, P)) = P$$

- 2.其中一个密钥可以是公共的，因为每个密钥只执行一半的E“+”D

RSA算法

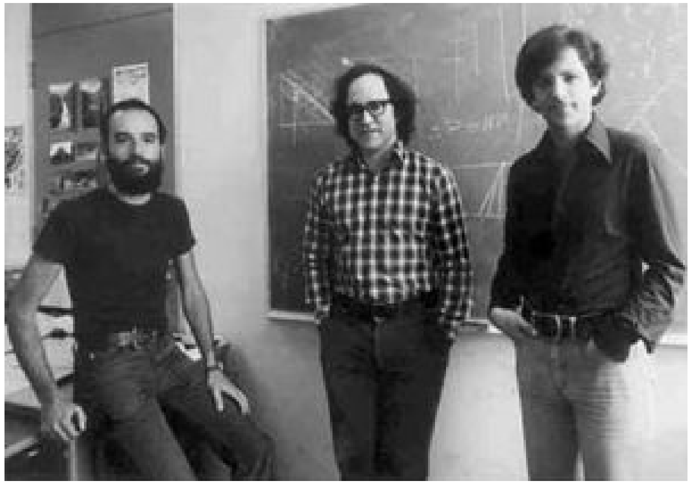
- DES及AES属于对称密码体制(symmetric cryptosystem)，加密及解密使用同一密钥。
- RSA算法是1977年由麻省理工学院的Rivest, Adi Shamir, Leonard Adleman提出的公钥密码体制(public-key cryptosystem)。
- 公钥密码体制也称非对称密码体制(asymmetric cryptosystem)。
- 在公钥密码体制中，加密密钥与解密密钥是不同的，加密密钥简称公钥(public key)，解密密钥简称私钥(private key)。

RSA算法

- 最著名、应用最广泛的公钥加密方案
- 基于有限域上模素数的整数的幂运算
  - 求幂需要 $O((\log n)^3)$ 运算（简单）
  - 使用大整数（例如1024位）
- 安全性来源于分解大整数因子的成本
- 因式分解需要 $O(e^{\log n \log\log n})$ 操作（困难）

RSA算法

- 图灵奖（2002年）：授予Ronald L. Rivest, Adi Shamir, Leonard M. Adleman图灵奖以表彰其使得公钥密码技术在实际中应用中的创造性贡献。
- RSA DATA SECURITY：1982年创立，2006年被EMC（1979年成立）收购。
- DNA电脑与生物电脑之父：1994年，美国南加州大学教授雷纳德·阿德勒曼（L.Adleman）博士，在《科学》杂志上发表一篇题为《组合问题的生物电脑解决方案》的论文，首次提出分子计算机，即用DNA分子构建电脑的设想。



RSA三杰（1977年）

RSA三杰

RSA算法

- 首先，选取两个大素数：p和q，计算乘积：n=p*q
- 其中n公开，p、q保密。
- 然后随机选取加密密钥e，使e和(p-1)*(q-1)互素。
- 接着要找出d，使得：
- e*d = 1 mod ((p-1)*(q-1))

RSA算法

- 加密与解密时都没有用到：p、q、(p-1)*(q-1)
- 只要n足够大，比如达到1024位，即2^{1024}，则在短时间内无法把n分解成p、q的乘积。

- 按下面的公式进行加密：

$$c=m^e(\mathrm{mod}\ n)$$

- 按下面的公式进行解密：

$$m=c^d(\mathrm{mod}\ n)$$

RSA算法-具体流程

1、随机选择两个不等素数p和q。

2、计算出n=p*q。

3、选择一个数e使它和(p-1)(q-1)互素。

4、计算e在模(p-1)(q-1)的逆元d。

5、公开 (e,n)作为RSA公钥。

6、保留 (d,n)作为RSA私钥。

RSA算法-举例

- 举例说明其工作过程：取两个素数

  p=11，q=13，n = p*q = 11*13 = 143，

  z=(p-1)*(q-1)=(11-1)*(13-1)= 120，

- 再选取与z=120互素的整数e，如e=7，现可计算出满足 7*d = 1 mod 120 的整数 d=103(私钥)，即：7*103=1 mod 120

- 整理如下：
  - p=11，q=13，n=143，
  - e=7，d=103，
  - (n,e)=(143,7)，
  - (n,d)=(143,103)

RSA算法-举例

- 以数据加密为例：
- A向B发送机密数据信息 m=85 ，并已知B的公钥 (n,e)=(143，7)，于是可计算出密文：

  $$c = m^e \bmod n = 85^7 \bmod 143 = 123$$

- A将c发送至B, B利用私钥(n,d)=(143,103)对c进行解密：

  $$m = c^d \bmod n = 123^{103} \bmod 143 = 85$$

RSA算法 -模幂运算

- 模幂运算可以使用模重复平方算法（Repeated square-and-multiply algorithm），这是一种快速、高效的求幂算法
- 将指数e用二进制形式表示，即$e = \sum_{i=1}^{r} e_i 2^i$，$e_i = 0$ 或 1
- 例：$7^5=7^4 \cdot 7^1=3 \cdot 7 = 10 \bmod 11$
- 例：$3^{129}=3^{128} \cdot 3^1=5 \cdot 3 = 4 \bmod 11$

RSA算法 -模幂运算

```
c = 0; f = 1
for i = k down to 0
    do c = 2 * c
       f = (f * f) mod n
    if b_i == 1 then
       c = c + 1
             f = (f * a) mod n
 return f
```

RSA算法的数学基础

- 数论四大定理：
  - 威尔逊定理
  - 费马小定理
  - 欧拉定理
  - 孙子定理（中国剩余定理）
- 其中在公钥密码学中起重要作用的两个定理是费马定理和欧拉定理。

RSA算法的数学基础 – Euler函数

- φ(n): 小于n且与n互素的整数个数。

- 例如φ(5) = 4, 因为与5互素的整数有: 1, 2, 3, 4
- 例如φ(10)= 4, 因为与10互素的整数有: 1, 3, 7, 9

Euler

RSA算法的数学基础 – Euler定理

- 若gcd(x,n)=1, 则$x^{\varphi(n)} = 1 \pmod{n}$

- 例如$3^{\varphi(5)} = 3^4 = 81 = 1 \pmod{5}$

RSA算法的数学基础 – Fermat小定理

- 设p为素数，且gcd(x,p)=1，则$x^{p-1} = 1 \pmod p$。
- 因为当p为素数时，显然有$\varphi(p) = p-1$。

Fermat

RSA算法的数学基础 － 中国剩余定理

- 数论最有用的结果之一是中国剩余定理（Chinese remainder theorem ，CRT）
- 中国数学家孙策在公元100年左右发现的
- 对于加快RSA公钥方案中的某些运算非常有用，允许计算模的模因子，然后把答案结合起来得到实际结果
- 由于计算成本与尺寸成正比，加速了模计算。

RSA算法的数学基础 - 中国剩余定理

- 设$m_1, m_2, m_3 \cdots, m_r$两两互素，则以下同余方程组

  $x \equiv a_i \pmod{m_i}$， $i=1,2,3,\cdots r$

  模 $M=m_1 m_2 m_3 \cdots m_r$ 的唯一解为
- $x = \sum_{i=1}^{r} a_i * M_i * (M_i^{-1} \bmod m_i) \bmod M$
- 其中$M_i = M/m_i$

RSA算法的数学基础 – 中国剩余定理

- 《孙子算经》

- 有物不知其数，三三数之剩二，五五数之剩三，七七数之剩二。问物几何？

RSA算法的数学基础 – 中国剩余定理

- 明朝数学家程大位在《算法统宗》中将解法编成易于上口的《孙子歌诀》

- 三人同行七十希，五树梅花廿一支，七子团圆正半月，除百零五便得知

- 将除以3得到的余数乘以70，将除以5得到的余数乘以21，将除以7得到的余数乘以15，全部加起来后再减去105或者105的整数倍，得到的数就是答案（除以105得到的余数则为最小答案）。

RSA算法的数学基础 – 中国剩余定理

例子 [编辑]

使用中国剩余定理来求解上面的“物不知数”问题，便可以理解《孙子歌诀》中的数字含义。这里的线性同余方程组是：

$$(S):\quad \begin{cases} x \equiv 2 \pmod{3} \\ x \equiv 3 \pmod{5} \\ x \equiv 2 \pmod{7} \end{cases}$$

三个模数$m_1=3$, $m_2=5$, $m_3=7$的乘积是$M=105$，对应的$M_1=35$, $M_2=21$, $M_3=15$. 而可以计算出相应的数论倒数：$t_1=2$，$t_2=1$，$t_3=1$. 所以《孙子歌诀》中的70，21和15其实是这个“物不知数”问题的基础解：

$$70 = 2 \times 35 \equiv \begin{cases} 1 \pmod{3} \\ 0 \pmod{5} \\ 0 \pmod{7} \end{cases},\ 21 = 1 \times 21 \equiv \begin{cases} 0 \pmod{3} \\ 1 \pmod{5} \\ 0 \pmod{7} \end{cases},\ 15 = 1 \times 15 \equiv \begin{cases} 0 \pmod{3} \\ 0 \pmod{5} \\ 1 \pmod{7} \end{cases},$$

而将原方程组中的余数相应地乘到这三个基础解上，再加起来，其和就是原方程组的解：

$$2 \times 70 + 3 \times 21 + 2 \times 15 \equiv \begin{cases} 2 \times 1 + 3 \times 0 + 2 \times 0 \equiv 2 \pmod{3} \\ 2 \times 0 + 3 \times 1 + 2 \times 0 \equiv 3 \pmod{5} \\ 2 \times 0 + 3 \times 0 + 2 \times 1 \equiv 2 \pmod{7} \end{cases},$$

这个和是233，实际上原方程组的通解公式为：

$$x = 233 + k \times 105,\ k \in \mathbb{Z}.$$

《孙子算经》中实际上给出了最小正整数解，也就是k=-2时的解：$x=23$.

RSA算法的数学基础 – 中国剩余定理 – 举例

- 例如韩信点兵问题：
- x = 1 mod 5　　(1)
- x = 5 mod 6　　(2)
- x = 4 mod 7　　(3)
- x = 10 mod 11　(4)

- 则M = 5*6*7*11 = 2310,
- M_1 = 6*7*11 = 462
- M_2 = 5*7*11 = 385
- M_3 = 5*6*11 = 330
- M_4 = 5*6*7 = 210

RSA算法的数学基础 – 中国剩余定理 – 举例

- $M_1^{-1} \bmod m_1 = 462^{-1} \bmod 5 = 3$
- $M_2^{-1} \bmod m_2 = 385^{-1} \bmod 6= 1$
- $M_3^{-1} \bmod m_3 = 330^{-1} \bmod 7= 1$
- $M_4^{-1} \bmod m_4 = 210^{-1} \bmod 11= 1$

- $\sum_{i=1}^{r} a_i * M_i * (Mi^{-1} \bmod mi) \bmod M$

=1*462*3 + 5*385*1 + 4*330*1 + 10*210*1 mod 2310

=2111