同余式

网络空间安全学院 胡丽琴

同余式

主要内容

一 一次同余式

二 中国剩余定理

三 RSA公钥密码系统

1 一次同余式

同余式

- 在代数里，一个很重要的问题就是解代数方程。在不定方程$ax+by=c(a,b\neq 0)$的求解过程中，如果引入同余的符号，则可以将它变成一个一次同余式，讨论起来将更加简单。

同余式

- **定义3.11** 设m是一个正整数，$f(x)$为多项式且

$$f(x)=a_n x^n+\dots+a_1 x+a_0,$$

其中a_i是整数，则

$$f(x)\equiv 0 \pmod{m} \tag{1.1}$$

叫做**模m同余式**。若$a_n \not\equiv 0 \pmod{m}$，则$n$叫做$f(x)$的次数，记为$\deg f$。此时，上式又叫做**模$m$的$n$次同余式**。

同余式

- 如果整数a使得

$$f(a)\equiv 0(\mathrm{mod}\ m)$$

成立，则a叫做同余式(1.1)的解。

- 如果已知存在整数a使得$f(a)\equiv 0(\mathrm{mod}\ m)$成立，$a$唯一吗？

- **例** $x^5+x+1\equiv 0(\mathrm{mod}\ 7)$是首项系数为1的模7的5次同余式，那么$a=2$是不是该同余式的解？

● 同余式

同余式解

- 由同余的性质可知，如果整数a是$f(x)\equiv 0 \pmod{m}$的解，则满足$x\equiv a \pmod{m}$的所有整数都是它的解。即：a所在剩余类C_a中的每个剩余都是该同余式的解。

- 同余式$f(x)\equiv 0 \pmod{m}$的**解数**定义为：**模m的完全剩余系**中使得同余式$f(x)\equiv 0 \pmod{m}$成立的剩余类个数。

● 同余式

- 求同余式$4x^2+12x-7\equiv 0 \pmod{15}$的解数.

- 解：取模15的绝对值最小完全剩余系：−7, −6，…，0，…，6，7。

- 将这些数代入同余式，计算$4x^2+12x-7 \pmod{15}$的值，验证是否为零。

- 通过验证可知，该同余式的解为$x\equiv -7,-2,-1,4 \pmod{15}$，所以解数为4.

● 同余式

- 求同余式$4x^2+12x-10\equiv 0 \pmod{15}$的解数.

> 解：取模15的绝对值最小完全剩余系：−7，−6，…，0，…，6，7。

> 将这些数代入同余式$4x^2+12x-10 \pmod{15}$

> 通过验证可知，该同余式的解为$x\equiv -5, 2, 5 \pmod{15}$，所以解数为3.

例

● 同余式

- 求同余式$4x^2+12x-3\equiv 0 \pmod{15}$的解数.
- 解：取模15的绝对值最小完全剩余系：−7，−6，…，0，…，6，7。
- 直接计算可知，所有整数都不是同余式$4x^2+12x-3\equiv 0 \pmod{15}$的解。
- 所以解数为0.

一次同余式

- 设$a(\mathrm{mod}\ m)\neq 0$，下面我们讨论最简单的同余式，一次同余式$ax\equiv b(\mathrm{mod}\ m)$的解。

- 判断同余式是否有解。
- 如果有解，给出求解方法及解。

一次同余式

- 假设一次同余式$ax\equiv b(\mathrm{mod}\ m)$有解，不妨设它的解为$x\equiv x_1(\mathrm{mod}\ m)$，即$ax_1\equiv b(\mathrm{mod}\ m)$
- 则$b=ax_1+my$，由此可以得到a，b和m之间的什么性质？或者说这三个参数之间满足什么条件？

一次同余式

- 如果一次同余式$ax \equiv b \pmod{m}$有解，则$(a, m) \mid b$.

- 那么，反之成不成立？即：如果$(a, m) \mid b$，一次同余式$ax \equiv b \pmod{m}$是不是一定有解？

- 设$(a,m)=1$，同余式$ax \equiv 1 \pmod{m}$有没有解？如果有解，有多少个解？解分别是什么？

一次同余式

定理

- **定理3.1.2** 设m是一个正整数，a是满足$(a,m)=1$的整数，则一次同余式

$$ax \equiv 1 \pmod{m} \tag{1.2}$$

有唯一解$x \equiv a' \pmod{m}$。

- **例** 求$3x \equiv 1 \pmod{7}$的解

● 一次同余式

- **定义3.1.1** 设m是一个正整数，a是一个整数，如果存在整数a'使得式

$$aa' \equiv 1 \pmod{m} \qquad (1.2)$$

成立，则a叫做**模m可逆元**。

- 由定理3.1.2，在模m意义下，a'是唯一存在的。这时a'叫做a的模m逆元，记作$a' \equiv a^{-1} \pmod{m}$。

● 一次同余式

- **例** 求$3x\equiv 2(\mathrm{mod}\ 7)$的解。

- 解：找出所有模7的绝对值最小完全剩余：-3，-2，-1，0，1，2，3，通过验证可知，$x\equiv 3(\mathrm{mod}\ 7)$是同余式$3x\equiv 2(\mathrm{mod}\ 7)$的解。

- 事实上，我们已知$3x\equiv 1(\mathrm{mod}\ 7)$的解为：$x\equiv 5(\mathrm{mod}\ 7)$

➢ 两个同余式的解有什么关系？

同余式

- **定理3.1.1** 设m是一个正整数，a是不整除m的整数，则一次同余式

$$ax \equiv b \pmod{m} \qquad (1.2)$$

有解的充分必要条件是$(a,m)|b$。而且当同余式(1.2)有解时，其解数为$d=(a,m)$。

● 一次同余式

- **例** 求解一次同余式$33x\equiv 22 \pmod{77}$.

- 解：$(33,77)=11$，$11|22$，所以同余式有解，且恰有11个解。

- 首先同余式各项系数除以11，得到同余式$3x\equiv 2 \pmod{7}$。
- 同余式$3x\equiv 1 \pmod{7}$的解为$x\equiv 5 \pmod{7}$.
- 从而同余式$3x\equiv 2 \pmod{7}$的解$x\equiv 2\times 5\equiv 3 \pmod{7}$.
- 最后得到同余式$33x\equiv 22 \pmod{77}$的解为$x\equiv 3+11t \pmod{77}$，其中$t=0,1,\ldots,10$。

● 一次同余式

- **定理3.1.3** 设m是一个正整数，a是满足$(a,m)|b$的整数，则一次同余式

$$ax \equiv b \pmod m$$

的全部解为

$$x = \frac{b}{(a,m)} \cdot \left(\left(\frac{a}{(a,m)}\right)^{-1} \left(\bmod \frac{m}{(a,m)}\right)\right) + t\frac{m}{(a,m)} \pmod m$$

$t=0,1,\ldots,(a,m)-1.$

● 一次同余式练习

- **定理3.1.4** 设m是一个正整数，则整数a是模m简化剩余的充要条件为整数a是模m可逆元。

- 求下列一次同余式的解。

  - $6x \equiv 3 \pmod 9$
  - $15x \equiv 9 \pmod{25}$

2

中国剩余定理

● 物不知其数

定理

- 当年黄蓉身中剧毒，郭靖将她送到瑛姑那里救治，进入瑛姑茅舍，瑛姑就给她出了一题：

“今有物不知其数，三三数之剩二，五五数之剩三，七七数之剩二。问物几何？”

- 大家知道这个数是多少吗？

中国剩余定理

- 其实，这就是享誉中外的“中国剩余定理”
- 中国剩余定理(孙子定理)就是“物不知其数”问题的推广形式。
- 剩余问题：在整数除法里，一个数同时除以几个数，均有剩余；已知各除数以及对应的余数，要求出满足条件的被除数问题，称为剩余问题。

物不知其数

- 古代人的解法：凡三三数之剩一，则置七十；凡五五数之剩一，则置二十一；凡七七数之剩一，则置十五；一百六以上，以一百零五减之即得。

- 译成算式解就是：

  - $70\times2+21\times3+15\times2=233$
  - $233-105\times2=23$.

● 基础数法

定理

- 现代人的解法
  - 用各除数的“基础数”求解
- 基础数的条件：
  - 此数必须符合除数自身的余数条件。
  - 此数必须是其它所有各除数的倍数。

● 基础数法

- “物不知其数”问题：
- 求各除数的最小公倍数：$3\times5\times7=105$
- 求各除数的基础数。
  - ✓ [3]：$105\div3=35$，$35\div3=11\ldots2$
  - ✓ [5]：$105\div5=21$，$21\div5=4\ldots1$，基数：21×3
  - ✓ [7]：$105\div7=15$，$15\div7=2\ldots1$，基数：15×2
- 各基数的和：$35+63+30=128$
- 适合条件的数x：$128-105=23$.

● 中国剩余定理

- **定理3.2.1(中国剩余定理)**设$m_1,\ldots,m_k$是k个两两互素的正整数，则对任意的整数$b_1,\ldots,b_k$，同余式组

$$\begin{cases} x \equiv b_1 \pmod{m_1} \\ \cdots\cdots\cdots\cdots \\ x \equiv b_k \pmod{m_k} \end{cases}$$

一定有解，且解是唯一的。事实上，若令

$$m=m_1\ldots m_k,\ m=m_iM_i,\ i=1,\ldots,k,$$

则同余式组(2.1)的解可表示为

$$x\equiv M_1'M_1b_1+\ldots+M_k'M_kb_k \pmod{m},$$

其中$M_i'M_i\equiv1 \pmod{m_i}$, $i=1,\ldots,k$.

中国剩余定理

- “物不知其数”问题，可转化为解下列一次同余式组：

$$x \equiv 2 \pmod 3$$

$$x \equiv 3 \pmod 5$$

$$x \equiv 2 \pmod 7$$

● 中国剩余定理

定理

- 由定理3.2.1得：$M_1=35$，$M_2=21$，$M_3=15$，通过广义Euclid除法可得，$M_1'=2$，$M_2'=1$，$M_3'=1$，所以，最终得到该同余式组的解为

$$x \equiv M_1M_1'b_1+M_2M_2'b_2+M_3M_3'b_3$$

$$=140+63+30=233\equiv 23 \pmod{105}.$$

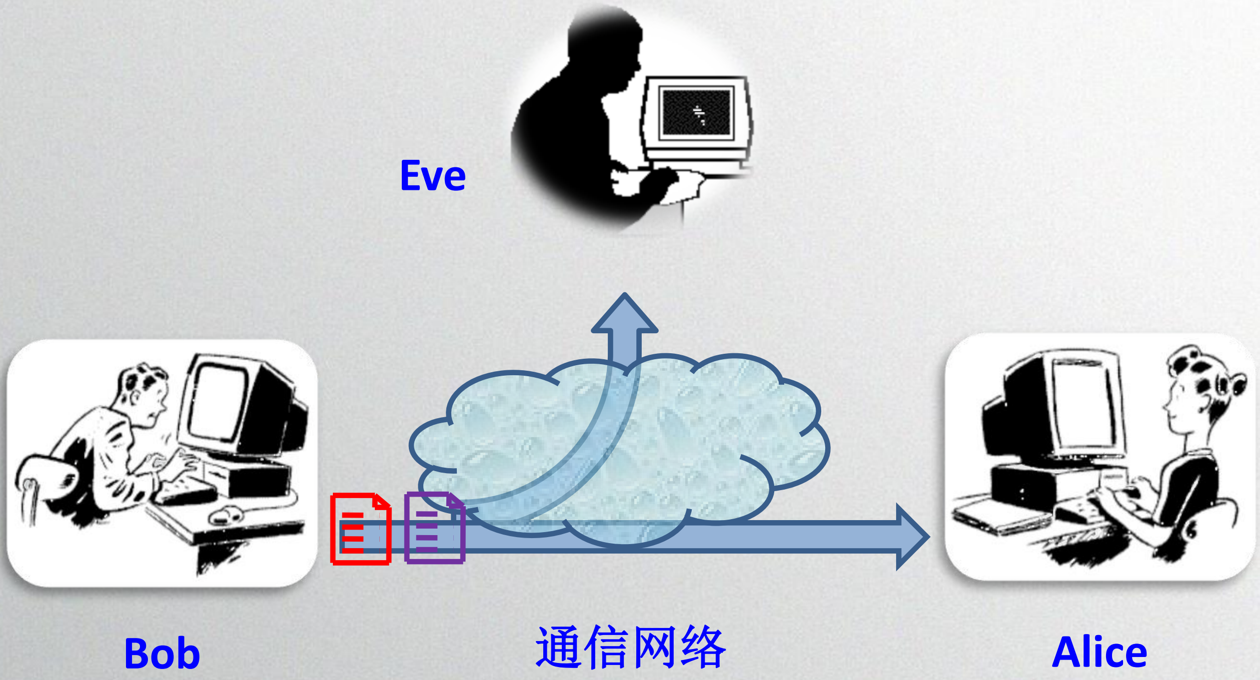
中国剩余定理

- **练习：** 分别用模重复平方计算法和中国剩余定理计算$3^{1000000} \pmod{77}$.

3

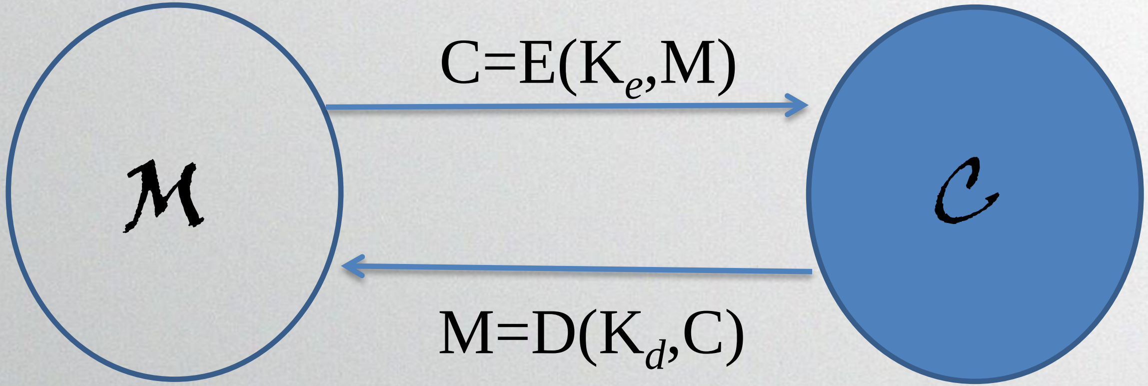
RSA公钥密码系统

通信模型

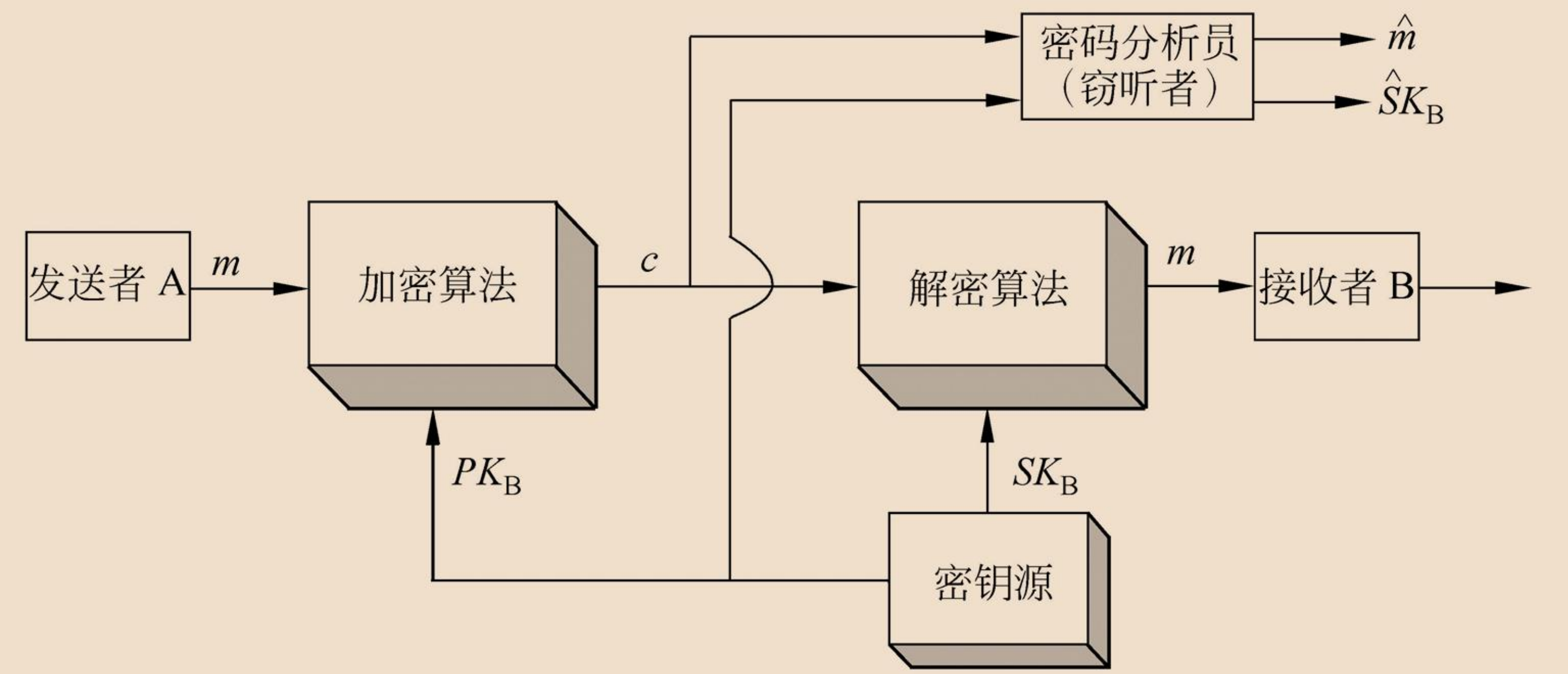


密码系统

- 密码系统图示

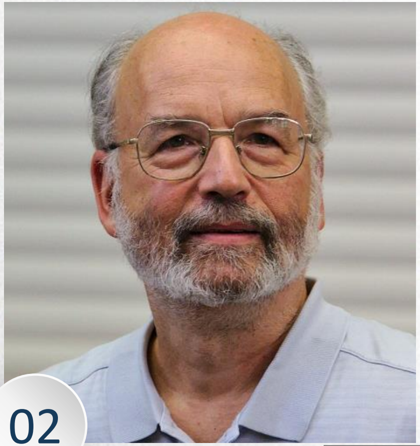


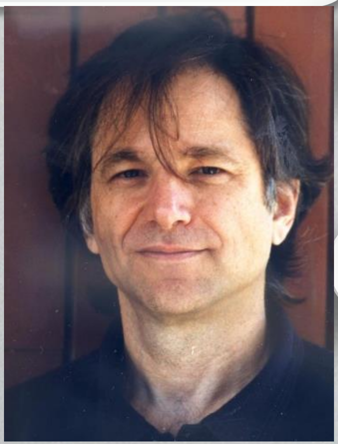
公钥体制加密框图



RSA公钥密码

RSA是第一个比较完善的公开密钥算法，它既能用于加密，也能用于数字签名。





01

02

03

RSA是以它的三个发明者Ron Rivest，Adi Shamir，Leonard Aldeman的名字首字母命名

●RSA体制

- RSA是最容易理解和实现的公钥密码算法。
- RSA算法是一种非对称密码算法(公钥密码算法)，所谓非对称，就是指该算法需要一对密钥，使用其中一个加密，则需要用另一个才能解密。
- RSA的理论基础是一种特殊的可逆模幂运算，其安全性基于分解大整数的困难性。

● 大数分解问题

- 例：设p=20000000000000002559，q=8000000000000001239，均为素数，求$n=pq$=?
- 已知 n=1600000000000000229500000000000003170601 是两个素数的乘积，求n的素因子分解。

● RSA公钥密码

- **定理3.3.1** 设p, q是两个不同的奇素数，$n=pq$。设整数e满足$1<e<\varphi(n)$，$(e,\varphi(n))=1$，那么
  - 存在整数d，$1\le d<\varphi(n)$，使得

$$ed\equiv 1(\mathrm{mod}\ \varphi(n));$$

  - 对于任意整数a，$(a, n)=1$，以及a^e模n的最小正余数c，即$c\equiv a^e(\mathrm{mod}\ n)$，则有

$$c^d\equiv a(\mathrm{mod}\ n).$$

● RSA体制

- 随机产生两个不同的大素数p, q，二者具有相同的阶
- 计算$n=p\times q$，$\varphi(n)=(p-1)\times(q-1)$
- 随机选取整数e: $1<e<\varphi(n)$，使得e与$\varphi(n)$互素
- 运用广义Euclid除法计算唯一的整数d: $1<d<\varphi(n)$，使得

$$e\times d\equiv 1 \pmod{\varphi(n)}$$

- 公钥：$K_e^A=(n, e)$
- 私钥：$K_d^A=(p, q, d)$.

● RSA公钥加密

- 发送方：
  - 获取接收方A的公钥$K_e^A=(n, e)$。(从认证中心、电话本或信息公告栏等处得到)
- 将明文信息表示为整数m，$0\le m\le n-1$，$(m,n)=1$。(要求以最有效的方式来表示信息。)
- 计算整数$c=K_e^A(m)\equiv m^e(\bmod n)$，$1\le m\le n-1$.
- 将整数c转化成密文信息
- 将密文信息发送给接收方A。

RSA公钥解密

- 接收方：
  - 将密文信息转换成整数c
  - 运用解密私钥K_d^A恢复整数

    $m=K_d^A(c)\equiv c^d(\mathrm{mod}\ n)$
  - 将整数m转换成明文信息。

● RSA体制

- 在RSA密钥生成、加密、解密算法中，不容易解决的就是密钥d的求解

- **例** 设p=3，q=11，则$n=p\times q=3\times 11=33$，$\varphi(n)=(p-1)(q-1)=2\times10=20$

- ➢ e=3，(3,20) =1，$e\times d = 1 \pmod{\varphi(n)}$即$3d=1 \pmod{20}$，那么$d$=?

●RSA体制

- **例** 设p=47，q=71，e=79，求用RSA算法加密的公钥和私钥。
- 计算如下：

(1) $n=pq=47\times71=3337$

(2) $\varphi(n)=(p-1)\times(q-1)=46\times70=3220$

(3) 选取 e=79

(4) 私钥d应该满足：$79\times d=1 \pmod{3220}$

$$79\times d = 1 \mod 3220$$

$$79\times d - 3220\times k = 1$$

d=1019

用3220对79取模到的余数60代替3220

k=25

$$79\times d - 60\times k = 1$$

用79对60取模到的余数19代替79

d=19

$$19\times d - 60\times k = 1$$

用60对19取模得到的余数3代替60

k=6

$$19\times d - 3\times k = 1$$

用19对3取模得到的余数1代替19

d=1

$$d - 3\times k = 1$$

k=0

当d的系数最后化为1时，令k=0

● RSA公钥密码

- **例** 设p=7，q=17，e=5，求用RSA算法加密的公钥和私钥。
- 计算如下：

(1) $n=pq=7\times17=119$

(2) $\varphi(n)=(p-1)\times(q-1)=6\times16=96$

(3) e=5，私钥d应该满足：$5\times d=1 \pmod{96}$

● RSA公钥密码

$$5 \times d \equiv 1 \pmod{96}$$

$$5 \times d - 96 \times k = 1$$

96 mol 5 = 1　用1代替96

$$5 \times d - k = 1$$

当k的系数最后化为1时，令d=1

$\boldsymbol{d = 1}$

$\boldsymbol{k = 4}$

$\boldsymbol{d = 77}$

● RSA体制

- 例 设p=11，q=13。
  - 密钥生成：$n=pq=11\times13=143$

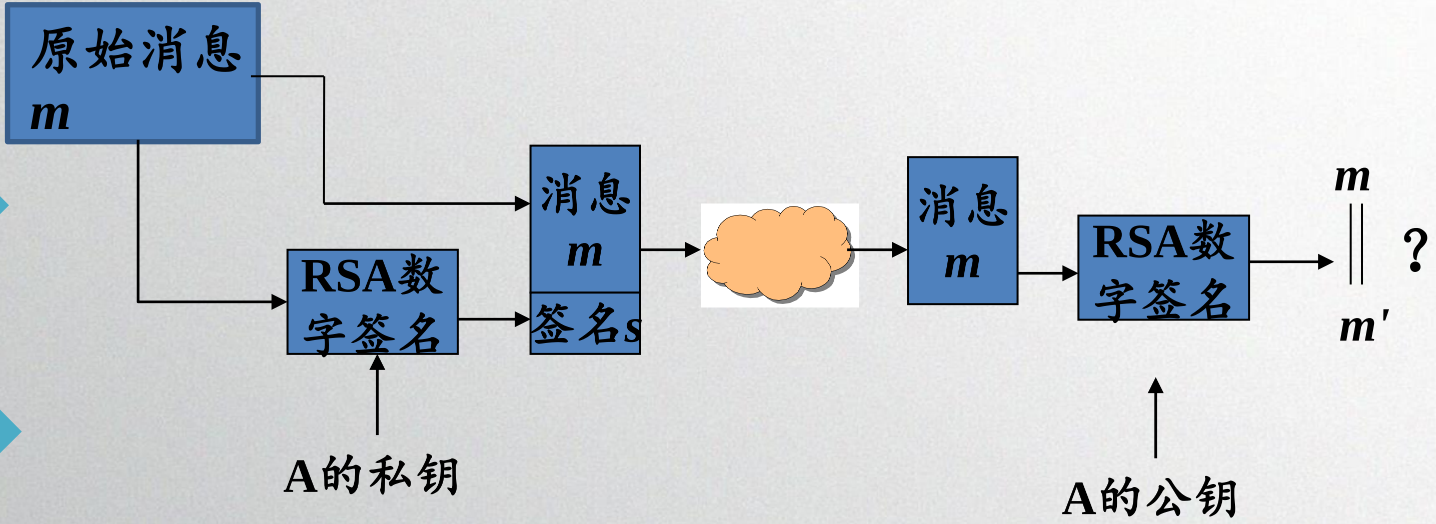
    $\varphi(n)=(p-1)\times(q-1)==(11-1)\times(13-1)=120$，

    选取e=17，通过$ed = 1 \ (\mathrm{mod}\ \varphi(n))$，计算得$d$=113。

    因此，公钥为$(n,e)=(143,17)$，私钥为d=113
  - 加密：用公钥e=17对明文m=24进行加密，则密文为：

    $c \equiv m^e (\mathrm{mod}\ n)$，即：$c \equiv 24^{17} (\mathrm{mod}\ 143)$
  - 解密：用私钥对密文进行解密：$m \equiv c^d \equiv 7^{113} \equiv 24 (\mathrm{mod}\ 143)$

● 数字签名



RSA数字签名

- 参数生成：生成公共参数N，配对的公钥e和私钥d
- 签名：消息m利用公共参数N和私钥d，产生签名s
- 验证：对于消息与签名对$(m||s)$，s利用公共参数N和公钥e，计算得到m'，若$m' = m$，验证通过；否则不通过

●RSA数字签名

- 在RSA密钥生成、加密、解密算法中，不容易解决的就是密钥d的求解
- **例** 设p=3，q=11，则$n=p\times q=3\times 11=33$，$\varphi(n)=(p-1)(q-1)=2\times 10=20$
- e=3，(3,20) =1，$e\times d = 1 \pmod{\varphi(n)}$即$3d$=1 (mod 20)，那么$d$=?

●RSA数字签名

■ 参数建立：

➢ 秘密选取两个大素数p与q，计算$N=pq$及$\varphi(N)=(p-1)(q-1)$；

➢ 随机选取正整数e：$1<e<\varphi(N)$使$\gcd(e,\varphi(N))=1$；

➢ 解同余方程$ex\equiv 1(\bmod\ \varphi(N))$求出正整数$d$；

➢ 将e作为公钥公开，d作为私钥保密。

● RSA数字签名

- 签名生成：
  - 对于消息$m \in Z_N^*$，A计算$s=m^d(mod\ N)$.
  - s是A对消息的签名，A将s和消息m同时发送给B。
- 签名验证：
  - B收到s后，从公开信道上获取A的公钥e和公开参数N；
  - B计算$m'=s^e(mod\ N)$，验证$m=m'$是否成立。若成立，则接受A的签名s，否则拒绝此签名。