公钥密码背景及数学基础

清华大学计算机系
于红波
2023年5月10日

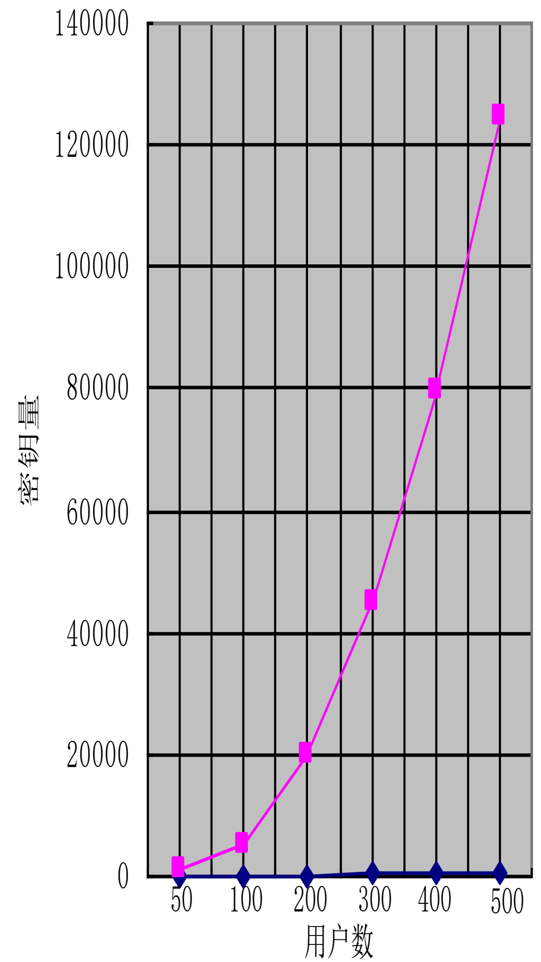
公开密钥密码算法

- 公钥密码体制产生的背景
- 公钥密码体制原理
- 公钥密码学基础

公钥密码体制产生背景

对称密钥加密的局限性

- 问题一：对称密钥分配困难
  - 对称密码体制在进行密钥分配时，要求通信双方已经有了一个共享的密钥，或者籍助于密钥分配中心分配会话密钥。
    - 完全依赖成本高的人工信使或者密钥分配中心的可靠性
  - 对称密码体制中，各个通讯方两两使用一对密钥，用户量增大时密钥空间急剧增大，密钥的更新和存储是难题。
    - n个用户需要C(n,2)= n(n-1)/2 个密钥
    - n= 100 时，C(100,2)= 4,995
    - n= 5000 时，C(5000,2)= 12,497,500



用户数与密钥量的对应关系

对称密钥加密的局限性

- 问题二：密钥的存储和保密

问题：密钥应该存储在哪里？

人脑

安全的房间

个人计算机

智能卡

对称密钥加密的局限性

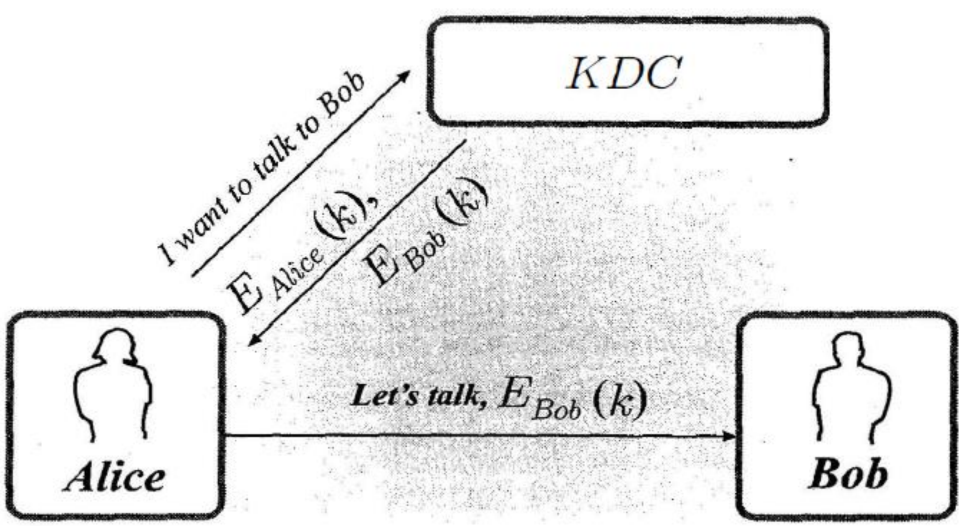
- 开放系统
  - 在“封闭”系统中安全通讯，使用物理的方法来分配密钥
  - 在开放系统中，通信方无法安全的分配密钥

如网上购物需要加密

给国外的同事写邮件需要加密

密钥分配中心

- 密钥分配中心 KDC
  - 所有人都和 KDC 共享一个密钥
  - 当 Alice 要和 Bob 通信时，首先请求 KDC 获取随机密钥，KDC 鉴别后将随机密钥分发给 Alice 要和 Bob



密钥分配中心

- KDC 介入的优点
  - 所有人只需要维护一个密钥
  - 系统开放性好
  - 密钥分配简单，存储容易
- 缺点
  - 所有通信的安全基于 KDC 的安全
  - KDC 通信负担重
- 解决方案
  - 使用密钥协商协议

公钥密码体制

- 1976年，斯坦福大学研究生W. Diffie和教授Martin Hellman提出了公钥密码体制，突破了对称密码体制中的难题。
- 公钥密码又称为双钥密码、非对称密码，标志性文献：
  - W.Diffie and M.E.Hellman,
    New Directions in Cryptography,
    IEEE Transaction on Information Theory,
    V.IT-22.No.6, Nov 1976, PP.644-654
- 从此，开启了公钥密码体制研究的序幕！

公钥密码体制被广泛应用

- 1977年，MIT的Ron Rivest，Adi Shamir和Len Adleman提出了一个可以真正用加解密数据的公钥密码算法RSA算法。
- 1985年，椭圆曲线学被提出，在椭圆曲线上可以建立一套公钥密码体制，即现在流行的ECC算法。
- 公钥密码体制应用：
  - 公钥加密
  - 数字签名
  - 密钥分配

公钥密码体制原理

公钥密码算法

- 对称密钥密码系统的缺陷
  - 密钥必须经过安全的信道分配
  - 无法用于数字签名
  - 密钥管理复杂，密钥的数量：$O(n^2)$
- 1976年，Whitfield Diffie和Martin Hellman在提出了非对称密钥密码，也称公钥密码。
- 公钥密码是密码学历史上唯一的一次真正的革命，它是基于数学函数而不是代换和置换。

公钥密码体制

- 公钥密码体制有6个组成部分
  - 明文：可读的信息，做为加密算法的输入
  - 加密算法：对明文进行的各种变换
  - 公钥/私钥：一个用于加密，一个用于解密；加密算法执行的变换依赖于公钥和私钥
  - 密文：加密算法的输出，不可读信息。密文依赖于明文和密钥，不同的密钥产生不同的密文
  - 解密算法：根据密文和相应的密钥，产生出明文

公钥密码算法的表示

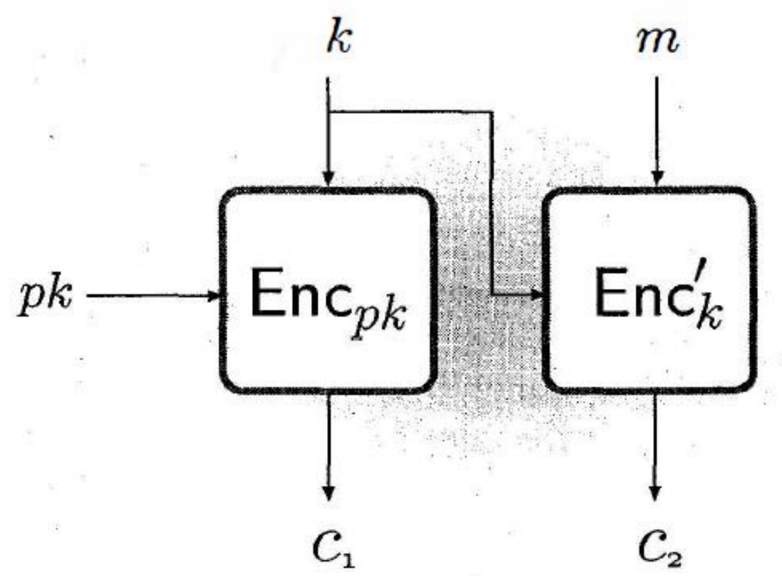
- 对称密钥算法
  - 密钥：会话密钥（Ks）
  - 加密函数：$C= E_{Ks}[P]$
  - 对密文C，解密函数：$D_{Ks}[C]$,
- 公开密钥算法
  - A: $(K_{Ua}$，$K_{Ra})$ 向B发送信息$((K_{Ub}$，$K_{Rb}))$:
  - 加密：$C= E_{KUb}[P]$, （用B的公开密钥加密）
  - 解密：$P= D_{KRb}[C]$
  - 签名：$E_{KRa}[P]$ （用A的私有密钥加密）
  - 验证：$D_{KUa}[C]$

公钥密码系统的应用

- 公钥密码系统有三种用途：
  - 加密/解密。
  - 数字签名：发送方用自己的私钥签署报文，接收方用对方的公钥验证对方的签名。
  - 密钥交换：双方协商会话密钥，用于对称密钥数据加密。

混合加密

- 混合加密
  - 发送方随机选择密钥 k，用接收方的公钥加密 k，得到密文 c_1，将 c_1 发给接收方
  - 双方协商了一个密钥 k，使用进行对称加密

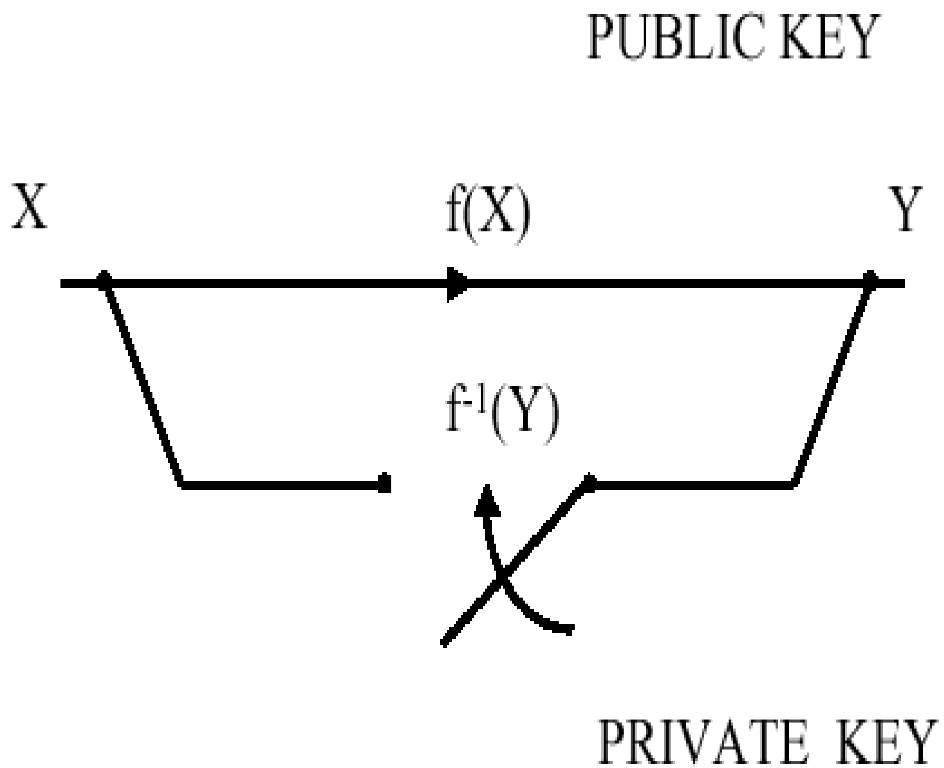


公钥密码算法应满足的要求

- 设计公钥密码的核心就是寻找合适的单向陷门函数
- 陷门单向函数是指该函数是易于计算的，但求它的逆是不可行的，除非再已知某些附加信息；当附加信息给定后，求逆可在多项式时间完成
- 单向陷门函数是一族可逆的满足下列条件的函数f：
  - (1)给定x，计算y=f(x)是容易的
  - (2)给定y，计算x使y=f(x)是困难的
  - (3)存在δ，已知δ时,对给定的任何y，若相应的x存在，则计算x使y=f(x)是容易的

Trap-door one-way function

Whitfield Diffie and Martin Hellman
"*New directions in cryptography*," 1976



素性检测和RSA算法

素性检测

- 确定性算法（deterministic algorithm）
  - 试除法：若小于$\sqrt{n}$的任意一个数可以整除n，那么n为合数。 效率低
  - AKS方法：2002年，Agrawal, Kayal, Saxena宣布找到第一个多项式时间的素性检测算法。“PRIMES is in P ”
- 概率算法（probabilistic algorithm）
  - Fermat测试：利用Fermat小定理测试
  - 平方根检验
  - Miller-Rabin素数测试
  - Euler-Jacobi测试

概率算法

- 若运用不同的参数或使用不用的方法不止一次地运行算法，提高成功率

1. **Fermat检验：** Fermat小定理: n是素数, n∤a, 则$a^{n-1} \equiv 1 \bmod n$

因此有：如果n是素数，则同余成立；但是同余成立并不能说明n一定是素数，也可能是复合数。定义下列过程为Fermat检验：

如果n是素数，$a^{n-1} \equiv 1 \bmod n$；

如果n是复合数，可能有$a^{n-1} \equiv 1 \bmod n$。

方法：不断地用不同的a检验，一旦出现同余不成立的情况，则说明n一定不是素数。检验过程就是不断求幂的过程，故复杂度同于计算指数算法的复杂度。

2. 平方根检验

- 模运算——模n
  - 如果n是素数，1的平方根是1或-1；
  - 如果n是复合数，1的平方根是1或-1，也可能是其他

（注意：在模运算中-1就是n-1）这就是平方根检验。

如果n是素数，则√1 mod n=±1；

如果n是复合数，则√1 mod n=±1和可能的其他值。

例1：n=8（复合数），1 mod n的平方根是多少？

$1^2=1 \bmod 8$， $(-1)^2=1 \bmod 8$， $3^2=1 \bmod 8$ ， $5^2=1 \bmod 8$

故有4个解：1，-1，3，5。

例2：n=7（素数），1 mod n的平方根是多少？只有1和-1.

$1^2=1 \bmod 7$, $(-1)^2=1 \bmod 7$, $2^2=4 \bmod 7$, $(-2)^2=4 \bmod 7$,
$3^2=2 \bmod 7$, $(-3)^2=2 \bmod 7$, 之后的4,5和6不用检测，因4=-3 mod 7, 5=-2 mod 7且6=-1 mod 7.

3. Miller-Rabin检验：

- Fermat检验和平方根检验优美的结合起来
- 求强伪素数（高概率）

在这种检验中，把n-1写作一个奇数m和2的幂的乘积：$n\text{-}1=m*2^k$，在基数a中的Fermat检验可以写成下图：

$$\boxed{a^{n-1}=a^{m\times 2^k}=[a^m]^{2^k}=[a^m]^{\underbrace{2^{2^{\cdot^{\cdot^{\cdot^{2}}}}}}_{K次}}}$$

这里我们不是在一步中计算$a^{n\text{-}1}\bmod n$是否通过

$$\left\{\begin{array}{l} a^m \equiv 1 \bmod n \quad 或 \\ a^{2^j m} \equiv -1 \bmod n, 0 \le j < k \end{array}\right. \quad 当 \gcd(a,n)=1$$

3. Miller-Rabin检验：

判断n是否是素数，n>3，安全参数t（确定成功率）

$$n-1=m\cdot 2^k, 2\nmid m$$

$For\ i=1\ to\ t,\ do$

选随机数$a, 2\le a\le n-2$

计算$y=a^r \bmod n$

若$y\ne 1$, 且$y\ne n-1$, 则

$j\leftarrow 1$

当$j\le k-1$且$y\ne n-1$, 执行

$y\leftarrow y^2 \bmod n$

若y=1，则返回n为合数

$j\leftarrow j+1$

若$y\ne n-1$, 则返回n为合数

返回n为素数

若$a^m=\pm 1$，则下一步中就是1，并一直保持1到通过Fermat检验，同时显然也通过了平方根检验，故这种情况中可以直接停止计算

把a^m平方，若结果为+1，则最终一定能通过Fermat检验，但是不能通过平方根检验，因为这一步中是1，而前一步中是±1之外的其他值，故说明n是复合数并停止检验。若结果为-1，也能最终通过Fermat检验，且能通过平方根检验，因为下一步中为1，故说明n是强伪素数并停止检验。若结果是±1之外的其他值，则暂时不能确定，进入下一次平方，循环K次。

素性检测

- 现在最受欢迎的素性检测
  - 把整除性检验和Miller-Rabin检验结合起来，即先进行整除性检验确保所选数不是明显的复合数，然后再进行Miller-Rabin检验。

数论基础：欧拉函数

- 欧拉函数ϕ(n)： n是正整数,ϕ(n) 是比n小且与n 互素的正整数个数。
- 欧拉函数的性质
  - φ(1)= 0
  - 如p为素数，则φ(p)= p-1
  - 当（m,n)=1时，φ(mn)= φ(m) φ(n)。
  - 如果p为素数，则$\varphi(p^e)= p^e - p^{e-1}$
- 当n可分解为素数乘积 $n = p_1^{e_1} p_2^{e_2} \cdots p_s^{e_s}$ 时，联合上述四条规则可求出φ(n)的值
- 只有当n可以分解为素数时才可以求出大复合数φ(n)，即**求φ(n)的困难性依赖于n的因数分解的困难性**，这一点非常重要。

$$\varphi(n) = (p_1^{e_1} - p_1^{e_1-1}) \times (p_2^{e_2} - p_1^{e_2-1}) \times \cdots \times (p_s^{e_s} - p_1^{e_s-1})$$

数论基础：欧拉定理

- 欧拉定理
  - 若整数m 和n 互素，则 $m^{\phi(n)+1} \equiv m \mod n$
    等价形式 $m^{\phi(n)} \equiv 1 \mod n$
- 例如：
  - m=3，n=10；ϕ(10)=4；$m^{\phi(n)}=3^4=81$；81 mod 10 = 1
  - 即：81≡ 1 mod 10 ； 3^{4+1} = 243 ≡ 3 mod 10
- 推论：
  - 给定两个素数p, q，p<>q, 两个整数 n, m，使得n=pq，0<m<n； 则对于任意整数k，下列关系成立：
    $m^{k\phi(n)+1} \equiv m \mod n$

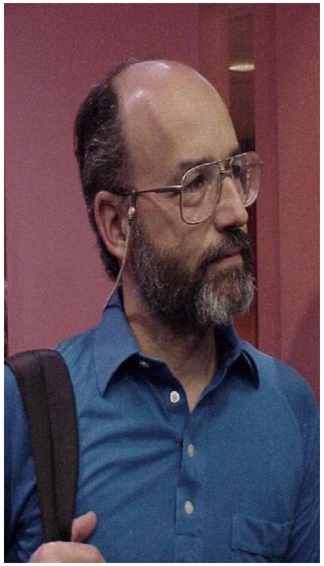
数学困难问题

- 因子分解问题（The Integer Factorization Problem, RSA体制）
- 二次剩余问题
- 离散对数问题：
  - 有限域的乘法群上的离散对数问题（The Discrete Logarithm Problem, ELGamal体制）
  - 定义在有限域的椭圆曲线上的离散对数问题（The Elliptic Curve Discrete Logarithm Problem，类比的ELGamal体制）

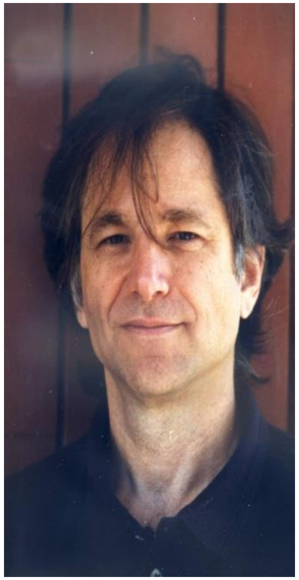
RSA算法

- 1978年， R.Rivest, A.Shamir和L.Adleman提出的一种用数论构造的公钥密码体制RSA算法，RSA算法已得到广泛的应用

Rivest



Shamir



Adleman

2002 Turing Award (June'03)

RSA算法简介

- MIT的Ron Rivest, Adi Shamir , Leonard Adleman于1977年提出1978年首次发表的RSA算法，是最早满足要求的公钥算法之一，被广泛接收而且被实现的通用公钥加密方法

- RSA体制是一种分组密码，其明文和密文都是0~n-1之间的整数，通常n的大小至少为1024位二进制数或者309位十进制数

RSA算法

- 密钥产生
  - 取两个大素数 p, q，保密；
  - 计算n=pq，公开n；
  - 计算欧拉函数$\phi(n) = (p-1)(q-1)$；
  - 任意取一个与$\phi(n)$ 互素的小整数e，即 $\gcd(e, \phi(n)) = 1$； $1 < e < \phi(n)$
  - 寻找d，$d < \phi(n)$，使得 $de \equiv 1 \bmod \phi(n)$，即$de = k\,\phi(n) + 1$
  - 公开 (e,n)
  - 将d 保密，丢弃p, q。
- 公开密钥:KU={e, n}
- 秘密密钥: KR={d, n}

设： p=7,q=17

则： n=119

$\Phi(n)=6\times16=96$

选择e=5

$5d=k\times96+1$

令 k=4

得到d=77

故知道:

KU = {5,119}

KR = {77,119}

RSA 算法加密/解密

- 利用： 公钥KU={e, n}和私钥KR={d, n}
- 加密过程
  - 把待加密的内容分成k比特的分组，k≤ $\log 2^n$，并写成数字，设为M，加密过程即为：$C = M^e \bmod n$
  - 例如：$C = M^5 \bmod 119$
- 解密过程
  - $M = C^d \bmod n$
  - 例如：$M = C^{77} \bmod 119$

RSA 算法的证明

- 试证明：解密过程是正确的。
- 证明：
  - $M = C^d \mod n$
  - $\quad = (M^e \mod n)^d \mod n$
  - $\quad = M^{ed} \mod n$
  - 即 $M^{ed} \equiv M \mod n$

- 根据欧拉定理推论: $M^{k\phi(n)+1} \equiv M \mod n$，得到 $ed = k\phi(n)+1$

RSA加密过程举例

- 取**p**=47，**q**＝71，
- 计算**n**=pq＝3337，**φ(n)**=(p-1)(q-1)＝3220，
- 随机选取正整数**d**=1019，d满足0<d<3220且gcd(1019, 3220)=1,
- 计算**e**=d^{-1}mod φ(n)=1019^{-1}mod 3220=79,
- 则公钥e＝79，n＝3337，私钥d＝1019,
- 对消息M=688加密：C=688^{79} mod 3337=1570,
- 解密：1570^{1079} mod 3337=688

RSA 算法的安全性

- 对RSA算法的攻击方法
  - 蛮力攻击：对所有密钥都进行尝试
  - 数学攻击：有多种数学攻击方法，他们的实质是两个素数乘积(n)的因子分解。
  - 计时攻击：这类方法依赖于解密算法的运行时间。
- 与其它密码体制一样，RSA抗穷举攻击的方法也是使用大密钥空间，但是密钥越大，系统运行速度越慢。

RSA 算法的安全性

- 大数的因子分解是数论中的一个难题。因子分解的进展可以用MIPS年描述计算的代价。
  - MIPS年是指一台每秒执行百万条指令的处理器运行一年。

十进制数字位数	近似比特数	完成日期	MIPS年
100	332	1991年4月	7
110	365	1992年4月	75
120	398	1993年6月	830
129	428	1994年4月	5000
130	431	1996年4月	1000
140	465	1999年2月	2000
155	512	1999年8月	8000

RSA 算法的性能

- 速度
  - 软件实现比DES 慢100倍
  - 硬件实现比DES慢1000倍

	512位	768位	1024位
加密	0.03	0.05	0.08
解密	0.16	0.48	0.93
签名	0.16	0.52	0.97
验证	0.02	0.07	0.08

RSA算法安全性分析

- 密码分析者攻击RSA体制的关键点在于如何分解n
- 若分解成功使n=pq，则可以算出φ(n)＝(p-1)(q-1)，然后由公开的e，解出秘密的d
- 若使RSA安全，p与q必为足够大的素数，使分析者没有办法在多项式时间内将n分解出来
  - 模n的比特数最小是1024，即n应当约为2^{1024}或309为十进制数位
  - 两素数p和q最小应该是512比特，约为154位十进制
  - 攻击进度：2020, RSA-250(829位)数也被成功分解。

RSA算法安全性分析

• 小加密指数问题

- 使用小加密指数e=3可以加快公钥加密或签名验证。
- 但若对同一消息m，使用小加密指数e=3同时发给不同人时，容易被攻击:
  - 设同时发给三个人，这三个人的模数分别为n_1, n_2, n_3，则有

$$c_1=m^3 \bmod n_1$$
$$c_2=m^3 \bmod n_2$$
$$c_3=m^3 \bmod n_3$$

  - 利用中国剩余定理可以解下列方程组

$$x \equiv c_1 \bmod n_1$$
$$x \equiv c_2 \bmod n_2$$
$$x \equiv c_3 \bmod n_3$$

  - 可以得到$x=m^3 \bmod N$，由此通过求三次方根可得到m。
- 解决这一攻击的方法是所谓的“消息加盐”，亦即在加密消息m时，针对不同的接收者，给m填入一些不同的随机比特。

RSA算法安全性分析

- **同模攻击**
  - 不同的人一定要使用不同的模数。
  - 假如使用同一模数n，则其中任何人都可以利用自己的公钥e和私钥d能分解n，从而能获得其他任何人的私钥。
  - 问题：若B与C具有相同的模数，用户A加密一消息x发送给B和C。A计算 $y_1 = x^{b_1} \bmod n, y_2 = x^{b_2} \bmod n$ ， 然后将y_1发送给B，y_2发送给C，假定O截取了y_1和y_2, O如何恢复出x?

$$c_1 b_1 + c_2 b_2 = 1$$

$$x = {y_1}^{c_1} {y_2}^{c_2} \bmod n$$

RSA算法安全性分析

- **素数多少问题**
  - 如果每个人都使用属于自己的不同的模数后，是否有足够的素数能满足这一要求。
  - 假如约定模长为1024比特位，则需要的素数大小为512比特位。由素数分布定理，512比特位的素数大约有：

$$2^{512}/\ln(2^{512}) = 2^{512}/512 = 2^{503}$$

个素数。由此可见，这样的素数是足够用的。

M比较小($m<2^{\ell}$)时对RSA攻击

Input: Public key $\langle N, e\rangle$; ciphertext c; parameter ℓ

Output: m such that $m^e = c \bmod N$

set $T := 2^{\alpha\ell}$

for $r = 1$ to T:

  $x_i := [c/r^e \bmod N]$

sort the pairs $\{(r, x_r)\}_{r=1}^{T}$ by their second component

for $s = 1$ to T:

  if $x_r = [s^e \bmod N]$ for some r

    return $[r \cdot s \bmod N]$

例： 当n=64, 则以大于0.35的概率存在长度最多为34比特的r,s, 使得m=rs

RSA PKCS#1v1.5

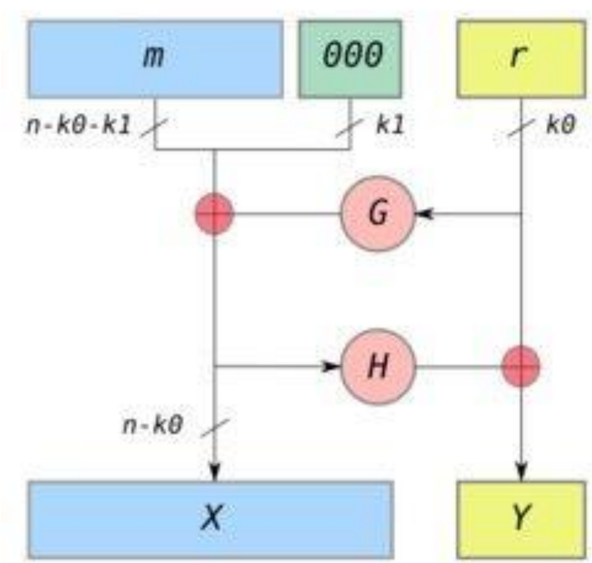
- 对于一个常见形式的公钥pk=<N,e>, 令k为N的比特长度，即k是一个满足$2^{8(k-1)} <= N < 2^{8k}$的整数。
- 假设要加密的消息m是一个8比特长的倍数，并且长度可以长达 k-11 字节。加密一个D比特的消息计算如下：
- $[(00000000 \| 00000010 \| r \| 00000000 \| m)^e \bmod N]$
- 其中r是随机生成的字符串，(k-D-3)字节，这些字节均不等于0。

RSA-OAEP（RSA PKCS#1 v2.0）

OAEP满足以下两个特性：

1． 添加了随机性的元素，可以用来将原有的确定性的加密方案（如传统的RSA加密方案）转换成为一种可能性的方案。

2． 防止对于密文的部分解密（或者其它的信息泄露），通过确保对手在不能反转陷门单向转换的情况下，无法复原明文的任何部分。CCA 安全的加密



OAEP机制：

G, H是两个Hash函数

RSA-OAEP加密方案：

密钥产生：产生公钥<N,e>和私钥<N,d>

加密：输入公钥<N,e>和消息$m \in \{0,1\}^l$

$m' := m \| 0^{k1}$然后选择一个随机数$r \in \{0,1\}^{k0}$。

然后计算

$X := m' \oplus G(r)$, $Y := r \oplus H(X)$

并设$M := X \| Y$ 。输出密文$c := [M^e \bmod N]$.

解密：输入私钥<N,d>和密文$c \in z_N^*$，计算$M := [c^d \bmod N]$.如果$\|M\| > l + k_0 + k_1$,输出⊥。否则，将M作为$X \| Y$。

计算$r := H(X)$和$M := G(r) \oplus X$。如果m’的所有的最低的k_1个bit不是0，输出⊥；否则，输出m’的最高的l bit。

DIFFIE-HELLMAN 密钥交换算法

Diffie-Hellman密钥交换算法

- 1976年，Diffie-Hellman第一个发表的公开密钥算法，被称为Diffie-Hellman密钥交换算法
- Diffie-Hellman密钥交换算法的目的是使两个用户能够安全地交换密钥；该算法本身也只局限于密钥交换
- Diffie-Hellman密钥交换算法的有效性在于计算离散对数非常困难

群

- 群：设G是一个非空集合，在G中定义了一个二元运算 ∘，若∘满足下面条件，则G称为一个群。

1. 任意 $a, b \in G$ ，则 $a \circ b \in G$
2. 对于任意的 $a, b, c \in G$, $(a \circ b) \circ c = a \circ (b \circ c)$

$\left.\right\}$ 半群

3. 在G中存在一个元素e，它对G中任何一个元素g，有

$$e \circ g = g \circ e = g$$

4. 对G中任何一个元素g，都存在一个元素g’，使得

$$g \circ g' = g' \circ g = e$$

e唯一，称为单位元

g’唯一，称为g的逆元

群

- 例：如整数集合Z对 数的加法 构成一个群；

  全体不等于0的有理数对普通数的乘法构成一个群；

  设 $N>1$ 为整数，模N 剩余类$Z_N=\{[k]\,|\,k\in Z\}=\{\overline{0},\overline{1},\ldots,\overline{N-1}\}$

  对模N 加法构成一个群。

  交换群：

  一个群如果对所有的 $a\circ b\in G$，都有 $a\circ b=b\circ a$，则称G是一个交换群（Abel）如加群。

  例：模N既约（简化）剩余类系：

  Z_N中与N互素的元素的集合：$Z_N^{*}=\{b\in\{1,2,\cdots,N-1\}\,|\,\gcd(b,N)=1\}$

  模N乘法运算：对任给 $a,b\in Z_N^{*}, a\times b=[a\times b \bmod N]$

  $(Z_N^{*},\times)$ 构成一个交换群，且Z_N^{*} 的阶为$\varphi(N)$

循环群

- 定义：若一个群G的每一个元素都是某一固定元素a的方幂， $G=\{a^n, n\in Z\}$则称G是循环群，也称G是由元素a生成的，记为G=<a>，a称为G的一个生成元。
- 例1：G=（Z，+）是一个循环群，G=<1>
- 例2:：设p是一个素数，则模p的简化剩余系 $(Z_p^*, \times)$ 构成一个循环群。模p的原根g为这个群的一个生成元

离散对数

- 对数是指数的反函数： $b = a^i$ => $i = \log_a b$
- 本原根（Primitive Root）
  - a是素数p 的一个本原根，如果a mod p, a^2 mod p , …, a^{p-1} mod p 是1到p-1的排列，即各不相同，是整数1到p-1的一个置换。

p= 19, a^i mod p, i=1,2,3,…,18																	
a	a^2	a^3	a^4	a^5	a^6	a^7	a^8	a^9	a^{10}	a^{11}	a^{12}	a^{13}	a^{14}	a^{15}	a^{16}	a^{17}	a^{18}
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	4	8	16	13	7	14	9	18	17	15	11	3	6	12	5	10	1
3	9	8	5	15	7	2	6	18	16	10	11	14	4	12	17	13	1
4	16	7	9	17	11	6	5	1	4	16	7	9	17	11	6	5	1
5	6	11	17	9	7	16	4	1	5	6	11	17	9	7	16	4	1

离散对数

- 对于整数b（b<p） 和素数p 的一个本原根a, 可以找到一个唯一的指数i，使得：$b \equiv a^i \bmod p$ ，其中$0 \le i \le (p-1)$称为 b 的 以a 为底 模p的离散对数或指数，记为$ind_{a,p}(b)$
  - $ind_{a,p}(1)=0$, 因为 $a^0 \bmod p = 1 \bmod p = 1$;
  - $ind_{a,p}(a)=1$ 因为 $a^1 \bmod p = a \bmod p = a$;
- 对于$b = a^x \bmod p$
  - 已知a, x, p，计算b 是容易的
  - 已知a, b, p，计算x 是非常困难的

CDH &DDH

计算Diffie-Hellman问题(CDH)

判定Diffie-Hellman问题(DDH)

- 选定循环群G以及一个生成元g ∈ G，给定两个元素 h_1 和 h_2，定义 $DH_g(h_1,h_2) = g^{\log_g h_1 \cdot \log_g h_2}$

  也就是若 $h_1 = g^x, h_2 = g^y$，则 $DH_g(h_1,h_2) = g^{x \cdot y} = h_1^{\,y} = h_2^{\,x}$

CDH问题：当随机选定 h_1, h_2，计算 $DH_g(h_1,h_2)$

DDH问题：对一个群中随机选择的元素 h_1, h_2，把一个随机的群元素和 $DH_g(h_1,h_2)$ 相区分。

Diffie-Hellman密钥协商

Diffie and Hellman Receive 2015 Turing Award, $1 million prize

Diffie-Hellman 密钥交换过程

全局公开的参数：
- q 是一个素数，
- g < q ， g是q 的一个本原根

- A 选择一个私有的a，
  a<q
- 计算公开的Y_A，
  $Y_A = g^a \bmod q$

Y_A Y_B

- B 选择一个私有的b，
  b< q
- 计算公开的Y_B，
  $Y_B = g^b \bmod q$

- A计算会话密钥
  $K=(Y_B)^a \bmod q$

- B计算会话密钥
  $K=(Y_A)^b \bmod q$

$E_K(m)$

举例

- 全局公开参数： q=97， a = 5 ； 5是97 的本原根
- A选择私钥 $X_A=36$， B 选择私钥 $X_B=58$
- A 计算公钥 $Y_A= 5^{36} \bmod 97 = 50$
- B 计算公钥 $Y_B= 5^{58} \bmod 97= 44$
- A 与 B 交换公开密钥
- A 计算会话密钥
  - $K = Y_B^{X_A} \bmod q = 44^{36} \bmod 97= 75$
- B 计算会话密钥
  - $K = Y_A^{X_B} \bmod q = 50^{58} \bmod 97= 75$

ElGamal密码体制

$\mathbb{Z}_p^*$ 上的 ElGamal 密码体制

设 p 是一个素数，使得 $(\mathbb{Z}_p^*,\cdot)$ 上的离散对数问题是难处理的，令 $\alpha \in \mathbb{Z}_p^*$ 是一个本原元。令 $\mathcal{P}=\mathbb{Z}_p^*, \mathcal{E}=\mathbb{Z}_p^* \times \mathbb{Z}_p^*$，定义

$$\mathcal{K}=\{(p,\alpha,a,\beta):\beta\equiv\alpha^a \pmod p\}$$

p,α,β 是公钥，a 是私钥。
对于 $K=(p,\alpha,a,\beta)$，以及一个（秘密）的随机数 $k\in\mathbb{Z}_{p-1}$，定义：

$$e_K(x,k)=(y_1,y_2)$$

其中

$$y_1=\alpha^k \mod p$$

而

$$y_2=x\beta^k \mod p$$

对 $y_1,y_2\in\mathbb{Z}_p^*$，定义

$$d_K(y_1,y_2)=y_2(y_1^a)^{-1} \mod p$$

ElGamal密码体制

在 ElGamal 密码体制中，加密运算是随机的，因为密文既依赖于明文 x，又依赖于 Alice 选择的随机数 k。所以，对于同一个明文，会有许多 ($p-1$ 个) 可能的密文。

Elgamal 密码体制的工作方式可以非正式地描述如下：明文 x 通过乘以 β^k “伪装”起来，产生 y_2。值 α^k 也作为密文的一部分传送。Bob 知道密钥 a，可以从 α^k 计算出 β^k。最后用 y_2 除以 β^k 除去伪装，得到 x。

ElGamal密码体制

下面这个简单例子能够说明在 ElGamal 密码体制中所进行的计算。

Example

设 $p=2579, \alpha=2$。α 是模 p 的本原元。令 $a=765$，所以

$$\beta = 2^{765} \mod 2579 = 949$$

假设现在 Alice 想要传送消息 $x=1299$ 给 Bob。比如 $k=853$ 是她选择的随机数，那么她计算：

$$\begin{aligned} y_1 &= 2^{853} \mod 2579 \\ &= 435 \end{aligned}$$

和

$$y_2 = 1299 \times 949^{853} \mod 2579$$

当 Bob 收到密文 $y=(435, 2396)$ 后，计算：

$$\begin{aligned} x &= 2396 \times (435^{765})^{-1} mod\ 2579 \\ &= 1299 \end{aligned}$$

这正是 Alice 加密过的明文。

二次剩余问题、Rabin密码算法、Goldwasser-Micali加密方案

二次剩余（素数模）

- 二次剩余（定义）：给定群G，如果存在一个$x \in G$满足$x^2=y$，则元素$y \in G$是一个二次剩余。

在 Z_p^* 的特例中，如果存在x满足$x^2=y \bmod p$，则 称y是一个二次剩余。

- 命题：设p＞2为素数，Z_p^* 中的每个二次剩余都有两个平方根。

素数模二次剩余

证明

- 设y∈$\mathbf{Z}_p^*$是一个二次剩余。则存在x∈ $\mathbf{Z}_p^*$满足x^2=y mod p。显然，$(-x)^2 = x^2 = y \bmod p$。此外，$-x \neq x \bmod p$：如果-x = x mod p，则2x = 0 mod p，意味着p|2x。因为p为素数，这意味着p|2（因为p＞2，所以不可能成立）或p|x（因为0＜x＜p，所以也不可能成立）。所以,[x mod p]和[-x mod p]是$\mathbf{Z}_p^*$中互不相同的元素，即y至少有两个平方根。

  若x’是y的平方根，则$x'^2=x^2$=ymodp,（x’-x)(x’+x)=0modp, 则p|(x-x’)或p|(x+x’), 所以，x’=x modp 或x’=-x mod p

素数模二次剩余

- Z_p^* 中恰好一半元素为二次剩余。
  - 设$sq_p: Z_p^* \to Z_p^*$ 为函数$sq_p(x) \stackrel{\text{def}}{=} [\, x^2 \bmod p]$。根据之前命题说明$sq_p$是一个‘二对一’函数。
- 模p二次剩余的集合用符号QR_p表示，二次非剩余用QNR_p表示。当p>2为素数时，$|QR_p|=|QNR_p|= \frac{|Z_p^*|}{2} = \frac{p-1}{2}$
- 定义：$J_p(x)$为x模p的雅可比符号（Legendre符号）

$$J_p(x) = \begin{cases} 0, & x = 0 \\ +1, & x\text{是一个关于}p\text{的二次剩余} \\ -1, & x\text{是一个关于}p\text{的二次非剩余} \end{cases}$$

素数模二次剩余

- 描述p>2为素数时 $\mathbf{Z}_p^*$ 中二次剩余的特征
  - 首先 $\mathbf{Z}_p^*$ 是一个阶为p-1的循环群。设g是 $\mathbf{Z}_p^*$ 的一个生成元。意味着

$$\mathbf{Z}_p^* = \left\{ g^0, g^1, g^2, \cdots, g^{\frac{p-1}{2}-1}, g^{\frac{p-1}{2}}, g^{\frac{p-1}{2}+1}, \cdots, g^{p\text{-}2} \right\}$$

  - 将上面列出的元素逐个平方，再将幂关于p-1取模，产生一个 $\mathbf{Z}_p^*$ 中所有二次剩余的列表：$QR_p = \{ g^0, g^2, g^4, \cdots, g^{p\text{-}3}, g^0, g^2, \cdots, g^{p-3} \}$
  - 可发现 $\mathbf{Z}_p^*$ 中的二次剩余恰好是一些可以写作g^i的元素，其中$i \in \{0,\cdots,p\text{-}2\}$为偶数。

素数模二次剩余

- 命题：设p＞2为素数，则 $J_p(x) = x^{\frac{p-1}{2}} \bmod p$
- 证明：设g为 Z_p^* 中的任意一个生成元。如果x是模p的二次剩余，对于某些偶数i，有$x = g^i$，记$i = 2j$，其中j为整数，可得

$$x^{\frac{p-1}{2}} = (g^{2j})^{\frac{p-1}{2}} = g^{(p-1)j} = (g^{p-1})^j = 1^j = 1 \bmod p$$

所以 $x^{\frac{p-1}{2}} = +1 = J_p(x) \bmod p$,得证。

当x是二次非剩余时， $x = g^{2j+1}$,

$$x^{\frac{p-1}{2}} = (g^{2j+1})^{\frac{p-1}{2}} = g^{(p-1)j} \cdot g^{\frac{p-1}{2}} = g^{\frac{p-1}{2}} \bmod p$$

$(g^{\frac{p-1}{2}})^2 = g^{p-1} = 1 \bmod p$，由于g是生成元，所以 $g^{\frac{p-1}{2}} = -1 \bmod p$

素数模二次剩余

- 算法：测试一个给定元素$x \in Z_p^*$是否为二次剩余。
  - 判定素数模二次剩余
  - 输入：素数p；元素$x \in Z_p^*$
  - 输出：$J_p(x)$（或者输出x是二次剩余或二次非剩余）
  - $b = [\ x^{\frac{p-1}{2}} \bmod p]$
  - If b = 1 return “二次剩余”
  - else return “二次非剩余”

素数模二次剩余

命题：设p>2为素数，且x,y∈ $\mathbf{Z}_p^*$，则

$$J_p(xy) = J_p(x)\cdot J_p(y)$$

证明 应用前面的命题：

$$J_p(xy) = (xy)^{\frac{p-1}{2}} = x^{\frac{p-1}{2}} g y^{\frac{p-1}{2}} = J_p(x) g J_p(y) \bmod p$$

因为$J_p(xy)$，$J_p(x)$，$J_p(y)$ =± 1，此等式在整数域上也满足。

推论： 设p＞2为素数，x,x′∈QR_p，且y,y′∈QNR_p。则：

（1）[xx′mod p]∈QR_p

（2）[yy′mod p]∈QR_p

（3）[xy mod p]∈QNR_p

合数模二次剩余

- 命题： 设N=pq，其中p，q为不同的素数，$y \in \mathbf{Z}_N^*$ 且满足y $\boxed{\leftrightarrow}$ (y_p,y_q)。则当且进当y_p是一个关于模p的二次剩余，且y_q是一个关于模q的二次剩余时，y是一个关于模N的二次剩余。
- 证明：如果y是关于模N的二次剩余，由定义可知存在一个$x \in \mathbf{Z}_N^*$ ，满足$x^2=y \bmod N$。设x $\boxed{\leftrightarrow}$ (x_p,x_q)。则

$$(y_p, y_q) \leftrightarrow y = x^2 \leftrightarrow (x_p, x_q)^2 = ([x_p^2 \bmod p],[x_q^2 \bmod q])$$

其中$(x_p,x_q)^2$中是 $\mathbf{Z}_p^* \times \mathbf{Z}_q^*$ 中的元素(x_p,x_q)的平方。因而得到

$$y_p = x_p^2 \bmod p, y_q = x_q^2 \bmod q$$

且y_p，y_q为二次剩余（对相应的模而言）。

合数模二次剩余

- 重要结论：每个二次剩余$y \in Z_N^*$恰好有4个平方根
- 证明：设$y \leftrightarrow (y_p, y_q)$为一个关于模N的二次剩余，设$y_p, y_q$分别为关于模p和q的二次剩余。则y的4个平方根在$Z_N^*$中对应于：

$$(x_p,x_q), (-x_p,x_q), (x_p,-x_q), (-x_p,-x_q)$$

上面每一个都是y的一个平方根，因为

$$(\pm x_p, \pm x_q)^2 = ([(\pm x_p)^2 \bmod p],[(\pm x_p)^2 \bmod q])$$

$$= ([x_p^2 \bmod p],[x_q^2 \bmod q]) = (y_p, y_q) \leftrightarrow y$$

其中，符号$(\cdot,\cdot)^2$表示群$Z_p \times Z_q$中元素的平方。因为x_p和$-x_p$是模p唯一（类似的x_q和$-x_q$模q唯一），则中国剩余定理确保了方程上式中4个元素每个都对应到Z_N^*中不同的元素

合数模二次剩余

- 例： 考虑 $\mathbf{Z}_{15}^{*}$。元素4关于模15是一个二次剩余，且有一个平方根为2。因为2 $\boxed{\leftrightarrow}$(2,2)，因此可得4的其他平方根为

（1）$(2,[-2 \bmod 3]) = (2,1) \boxed{\leftrightarrow} 7$;

（2）$([-2 \bmod 5],2) = (3,2) \boxed{\leftrightarrow} 8$;

（3） $([-2 \bmod 5], ,[-2 \bmod 3]) = (3,1) \boxed{\leftrightarrow} 13$

可以验证$7^{2} = 8^{2} = 13^{2} = 4 \bmod 15$

合数模二次剩余

- 设QR_N表示模N二次剩余的集合。因为模N平方是一个4对1的函数的函数，因此立即可以得出 $\mathbf{Z}_N^*$ 中1/4的元素为二次剩余。可能注意到当且仅当y_p,y_q为二次剩余时， $y \in \mathbf{Z}_N^*$ 是一个二次剩余，所以QR_N和$QR_p \times QR_q$之间有一一对应的关系。因此，关于模N的二次剩余的比例为

$$\frac{|QR_N|}{\mathbf{Z}_N^*} = \frac{|QR_p| \cdot |QR_q|}{\mathbf{Z}_N^*} = \frac{\frac{p-1}{2} \cdot \frac{q-1}{2}}{(p-1)(q-1)} = \frac{1}{4}$$

定义扩展到N = pq为两个不同奇素数乘积时的情形。对于任何与N = pq互素的x有

$$J_N(x) \overset{\text{def}}{=\!=} J_p(x) \cdot J_q(x) = J_p([x \bmod p]) \cdot J_q([x \bmod q])$$

定义J_N^{+1}为$\mathbf{Z}_N^*$ 中的雅可比符号为+1元素的集合，类似地对 J_N^{-1}进行定义。

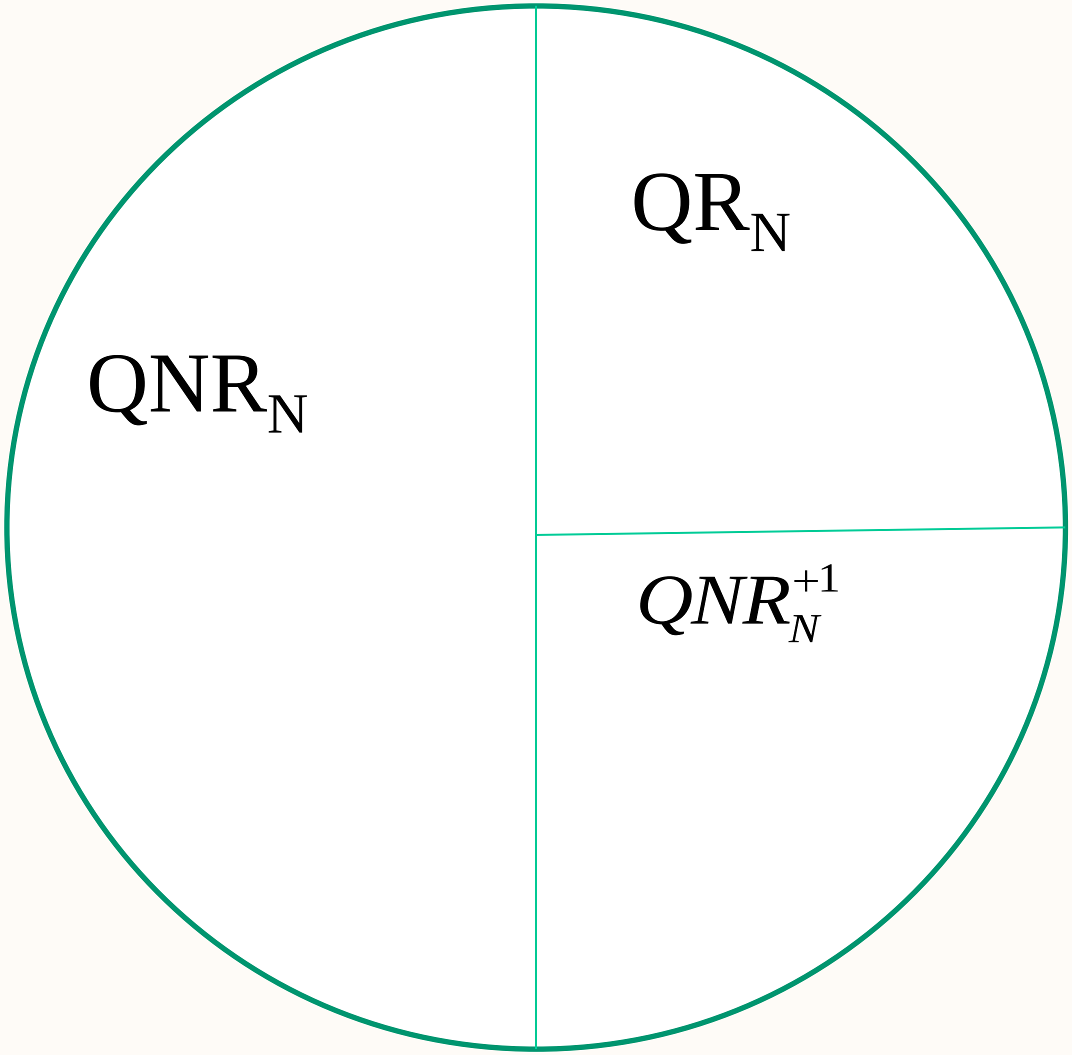
合数模二次剩余

- 如果X是一个关于模N的二次剩余，$J_N(x) = +1$。当$J_p(x) = J_q(x) = -1$时也可出现$J_N(x) = +1$。让符号QNR_N^{+1} 表示这类元素的集合

$$QNR_N^{+1} \underline{\underline{\mathrm{def}}} \left\{ x \in \mathbf{Z}_N^* \mid x\text{不是模}N\text{的二次剩余，但}J_N(x)=+1 \right\}$$

- 命题：　设N = pq，p,q为不相同的奇素数，则：
  （1）$\mathbf{Z}_N^*$ 中恰好一半元素属于 J_N^{+1}
  （2）QR_N包含于 J_N^{+1}
  （3）J_N^{+1} 中恰好一半元素属于QR_N（另一半属于QNR_N^{+1}　）

合数模二次剩余



合数模二次剩余

- 命题： 设N = pq为两个不相等奇素数的乘积，且x,y $\in \mathbf{Z}_N^*$ ，则有$J_N(xy) = J_N(x) \cdot J_N(y)$

证明

$J_N(xy)= J_p(xy)\cdot J_q(xy)=J_p(x)\cdot J_p(y)\cdot J_q(x) \cdot J_q(y)$

$=J_N(x) \cdot J_N(y)$

- 推论： 设N = pq为两个不同奇素数的乘积，且x,x′$\in QR_N$，y,y′$\in QNR_N^{+1}$ 。则有：

（1）$[xx' \bmod N] \in QR_N$

（2）$[yy' \bmod N] \in QR_N$

（3）$[xy \bmod N] \in QNR_N^{+1}$

合数模二次剩余

证明：只证明最后一个，其他的证明与其类似。因为$x \in QR_N$，可得$J_p(x) = J_q(x) = +1$。因为$y \in QNR_N^{+1}$可得$J_p(y) = J_q(y) = -1$。则

$J_p(xy)= J_p(x)\cdot J_p(y)=-1$和$J_q(xy)= J_q(x)\cdot J_q(y)=-1$

所以$J_N(xy)=+1$。但是xy不是一个关于模N的二次剩余，因为$J_p(xy)= -1$，所以[xy mod p] 不是模p的二次剩余。所以可以得到$xy \in QNR_N^{+1}$

合数模二次剩余-Jacobi 符号的性质

- 性质 1： N 是不同奇素数的乘积，则有

  $J_N(1) = 1 \quad J_N(-1) = (-1)^{\frac{N-1}{2}} \quad J_N(2) = (-1)^{\frac{N^2-1}{8}}$

- 二次互反律： 设P、Q是奇数，满足 $P > 1, Q > 1, (P, Q) = 1$,则有$J_P(Q)^* J_Q(P) = (-1)^{\frac{P-1}{2}\frac{Q-1}{2}}$。

- 例：计算 $J_{317}(105) = J_{105}(317) = J_{105}(2) = 1$

二次剩余假设

- 算法：
  - 已知合数的分解形式，判定某数是否为该合数模的二次剩余。
  - 输入：合数$N = pq$；因数p和q；元素 $x \in \mathbf{Z}_N^*$
  - 输出：一个判定结果，即是否$x \in \mathrm{QR}_N$
  - 计算 $J_p(x)$和 $J_q(x)$
  - If $J_p(x)= J_q(x)= +1$，return“二次剩余”
  - Else return “二次非剩余”

二次剩余假设

- 当N分解形式未知时，没有多项式算法可以判定x是否为模N的二次剩余。
- $J_N(x)$可以通过多项式时间求解。
- 部分测试：给定x, 当$J_N(x)=-1$，则x不可能是二次剩余。
- $J_N(x)=1$，不存在多项式在此情况下可以判定二次剩余。

计算模平方根

- 当p是一个奇素数，p模4余3. 设a 是模p的二次剩余， 则计算a模p的一个平方根的方法是

$$x := a^{\frac{p+1}{4}} \bmod p$$

证明：

$$J_p(a) = 1 = a^{\frac{p-1}{2}} \bmod p$$

$$a = a^{\frac{p-1}{2}+1} \bmod p = a^{2i+2} = (a^{i+1})^2 \bmod p$$

$$a^{i+1} = a^{\frac{p+1}{4}} \bmod p$$

Rabin算法

- Rabin算法安全性基于求合数的模平方根的难度，该问题等价于因子分解的难度。

1. Gen: 选取两个素数p,q, 两个都同余3模4。p,q作为私钥，N=pq作为公钥
2. Enc: 加密一个消息M<n 时，C=M^{2}mod N
3. Dec:

$$m_1 = C^{\frac{p+1}{4}} \bmod p$$

$$m_2 = (p - C^{\frac{p+1}{4}}) \bmod p$$

$$m_3 = C^{\frac{q+1}{4}} \bmod q$$

$$m_2 = (p - C^{\frac{q+1}{4}}) \bmod q$$

$$Let \quad a = q(q^{-1} \bmod p),\ b = p(p^{-1} \bmod q)$$

$$M_1 = (am_1 + bm_3) \bmod n$$

$$M_2 = (am_1 + bm_4) \bmod n$$

$$M_3 = (am_2 + bm_3) \bmod n$$

$$M_4 = (am_2 + bm_4) \bmod n$$

中国剩余定理（CRT）

假设$m_1,m_2,\ldots,m_n$两两互素，则对于任意的整数$a_1,a_2,\ldots,a_n$, 方程组

$$\begin{cases} x \equiv a_1 \bmod m_1 \\ x \equiv a_2 \bmod m_2 \\ \cdots \\ x \equiv a_n \bmod m_n \end{cases}$$

都存在整数解：

$$x = \sum_{i=1}^{n} a_i \times \frac{N}{m_i} \times \left[\left(\frac{N}{m_i}\right)^{-1}\right]_{m_i} \pmod N$$

其中 $N = \prod_{i=1}^{n} m_i$

Goldwasser-Micali加密方案

- 基于判定性二次剩余问题，等价于因子分解问题
  - Gen：输入1^n，选取两个素数p,q, N=pq，随机选择 $z \leftarrow QNR_N^{+1}$。公钥 $pk = \langle N, z\rangle$，私钥 $sk = \langle p, q\rangle$
  - Enc：输入公钥 pk 和消息 $m \in \{0,1\}$，随机选择 $x \leftarrow Z_N^*$，输出密文 $c = z^m \cdot x^2 \bmod N$
  - Dec：输入私钥 sk 和密文 c，判断 c 是否为模 N 的二次剩余，如果是输出0，否则输出1

谢谢！