整数分解

2024 年 10 月 24 日

1 实验目的

1. 理解并掌握整数分解的基本原理，掌握常用的整数分解方法，并且比较各方法的优缺点
2. 编程实现至少 3 种整数分解方法，分析各方法的优缺点

2 实验原理

- 问题描述

整数分解问题（算术基本定理的算法化）研究具有重要意义。RSA 是著名的公钥密码算法, 1977 年由 Rivest, Shamir 和 Adleman 一起提出, 在实际中有着广泛的应用. RSA 算法的安全性依赖于大整数分解困难, 这使得整数分解问题成为现代密码学非常关注的问题之一. 关于整数分解方法的综述可以参见 [1]. 整数分解为给定正整数 N, 求 N 的素因子, 即求整除 N 的素数

2.1 试除法

设 N 是一个正整数, 试除法看成是用小于等于 $\sqrt(N)$ 的每个素数去试除待分解的整数. 如果找到一个数能够整除除尽, 这个数就是待分解整数的因子. 试除法是最初等的整数分解算法, 该方法思想简单, 但能够快速分解出 N 中的小素因子

2.2 Pollard ρ 算法

Pollard 于 1975 年提出著名的 ρ 算法. 该方法的基本思想是通过多项式迭代产生数列, 从中寻找整数 x_1, x_2 满足 $\gcd(x_1 - x_2, N)$ 是 N 的一个非平凡因子

- 设 p 是 N 的一个素因子，$p < \sqrt{N}$. 找到两个整数 x, x', 满足 $x \equiv x' \pmod p$，则可通过求最大公因子 $\gcd(x - x', N)$ 来分解 N。

- 如果只是想随机的选一个子集 X 来进行碰撞。那么这个子集的大小大约要 $1.17\sqrt{p}$, 此时碰撞概率为 50%

- 但我们事先不知道 p 是多少，所以要计算 $\binom{|X|}{2} > p/2$ 次 GCD.

Pollard ρ 算法:

◇ 事实上，我们可以不那么随机。选取一个整系数多项式 $f(x)$, 例如 $f(x) = x^2 + 1$.

- ◇ 随机选取 x_1. 考虑序列 $x_1, x_2, \cdots$, 其中

$$x_k \equiv f(x_{k-1}) (\bmod N)$$

- ◇ 某一时刻必有

$$x_i \equiv x_j (\bmod p),$$

则

$$x_{i+1} \equiv f(x_i) \equiv f(x_j) \equiv x_{j+1} (\bmod p).$$

- ◇ 一般地，

$$x_{i+k} \equiv x_{j+k} (\bmod p)$$

- ◇ 更一般的，令 $j - i = \ell$, 只要 $j' \equiv i' (\bmod \ell)$, 则

$$x_{i'} \equiv x_{j'} (\bmod p)$$

- 这个算法需要计算 $\binom{j}{2}$ 次 GCD

进一步改进:

- 事实上在 $x_i, x_{i+1}, \cdots, x_j$ 中一定存在 $x_{i'}$ 使得

$$x_{2i'} \equiv x_{i'} (\bmod p)$$

- 这是因为连续 ℓ 个数一定有一个 ℓ 的倍数，设 $i \leq i' \leq j$ 满足 $i' = s\ell$.
- 此时 $x_{2i'} \equiv x_{i'} (\bmod p)$
- 这样最多 j 次就可以找到 i'.

算法分析:

- 从期望上来说 j 大概差不多是 $\sqrt{p}$ (生日碰撞)
- 而 $p < \sqrt{N}$，所以整个计算复杂度 $O(N^{\frac{1}{4}})$
- 算法有可能会失败。即 $x_i \equiv x_j (\bmod p)$ 会导致 $x_i \equiv x_j (\bmod N)$. 这个概率差不多是 p/N

例子 2.1. 设 $n = 7171$, $f(x) = x^2 + 1$, $x_1 = 1$ 则有数列:

$1, 2, 5, 26, 677, 6557, 4105, 6347, 4903, 2218, 219, 4936, 4210, 4560, 4872, 375, 4377, 4389, 2016, 5471, 88, 6 \cdots$

计算 $\gcd(x_1 - x_2\ n)$, $\gcd(x_2 - x_4, n), \gcd(x_3 - x_6, n) \cdots$ 发现 $\gcd(x_{11} - x_{22}, n) = 71$，这就找到了 n 的一个因子 71。

2.3 Pollard $p-1$ 算法

1974 年, Pollard 基于费马小定理, 提出了 $p-1$ 算法. 虽然该方法不是一个具有一般性的分解算法, 但是其思想却被应用到一些现代的分解算法中. 比如, 基于 Pollard $p-1$ 算法的思想, Lenstra 提出了椭圆曲线分解方法

Pollard $p-1$ 算法:

- 设 p 是 n 的一个素因子，并且假定对任意素数幂 $q \mid (p-1)$, 有
$$q \leq B.$$
- 于是
$$(p-1) \mid B!.$$
- $a \equiv 2^{B!} (\bmod\ n)$
- $a \equiv 2^{B!} \equiv 2^{p-1} \equiv 1 (\bmod\ p)$
- 此时 $p \mid \gcd(a-1, n)$.

例子 2.2. 假设 $n = 15770708441$. 我们选取 $B = 180$, 则可计算
$$a \equiv 2^{B!} \equiv 11620221425 (\bmod\ n)$$
于是 $\gcd(a-1, n) = 135979$.
$$15770708441 = 135979 \times 115979.$$

在这个例子中，135978 只有小素因子：
$$135978 = 2 \times 3 \times 131 \times 173.$$

Pollard $p-1$ 算法的缺陷：

- n 有一个素因子 p 使得 $p-1$ 只有小的素因子。
- $n = pq$, 其中 $p = 2p' + 1$, $q = 2q' + 1$, 且 p', q' 均为大素数。

2.4 其它算法:

费马曾提出一个基于二次同余的想法, 即如果可以找到正整数 s, t 满足 $s^2 \equiv t^2 (\bmod\ N)$, 则 $\gcd(s \pm t, N)$ 可能是 N 的一个非平凡因子. 例如 $10^2 \equiv 32^2 (\bmod\ 77)$, 通过计算 $(10+32, 77) = 7$, 得到 77 的一个因子 7.

为了提高搜索满足条件的整数 s, t 的效率, 人们引入了分解基的概念. 一个分解基是不超过某个上界 B 的素数集合, 若一个正整数的素因子均在该分解基中, 则称为 B-光滑的. 现代分解算法大都基于分解基的方法, 比如连分数方法, 二次筛法和数域筛法等. 数域筛法是目前分解大整数最有效的算法.

分解基方法的基本思想:

- 找到几个整数 x 使得 $x^2 (\bmod N)$ 是 B-光滑的
- 将某些 x 相乘使得每一个在分解基中的素数出现偶数次
- 这样就可以建立 $s^2 \equiv t^2 (\bmod\ N)$

例子 2.3. 假设 $n=15770708441$. 取 $b=6$，则 $\mathcal{B}=\{2,3,5,7,11,13\}$.

$$\begin{aligned} 8340934156^2 &\equiv 3\times 7(\mathrm{mod} n)\\ 12044942944^2 &\equiv 2\times 7\times 13(\mathrm{mod} n)\\ 2773700011^2 &\equiv 2\times 3\times 13(\mathrm{mod} n). \end{aligned}$$

于是

$$(8340934156\times 12044942944\times 2773700011)^2\equiv(2\times 3\times 7\times 13)^2(\mathrm{mod} n)$$

化简后

$$9503435785^2\equiv 546^2(\mathrm{mod} n).$$

从而我们有

$$\gcd(9503435785-546,15770708441)=115759$$

finding the factor 115759 of n.

假设 $\mathcal{B}=\{p_1,\ldots,p_b\}$ 是分解基. 令 c 是比 b 稍微大一点的数 (比如 $c=b+4$), 如果我们得到了 c 个同余式:

$$z_j{}^2\equiv p_1{}^{\alpha_{1j}}\times p_2{}^{\alpha_{2j}}\cdots\times p_b{}^{\alpha_{bj}}(\mathrm{mod} n)$$

$1\le j\le c$. For each j, consider the vector

$$a_j=(\alpha_{1j}\bmod 2,\ldots,\alpha_{bj}\bmod 2)\in(\mathbb{Z}_2)^b.$$

如果我们能找到一些 a_j 它们加起来模 2 是零向量 $(0,\ldots,0)$, 那么对应的 z_j 的乘积的分解就正好都是偶次幂的。

算法分析:

- $x^2\equiv y^2(\mathrm{mod}\ N)$ 导致 $x\equiv\pm y(\mathrm{mod}\ N)$. 当 N 至少有两个素因子时，这个发生的概率小于 1/2.
- 怎么选取这些 x 使得 $x^2(\mathrm{mod}\ N)$ 是 B-光滑的?
- 常用的方法: 可以选取这样的 $x=\sqrt{kN}$, $k=1,2,3,\cdots$。这样 $x^2(\mathrm{mod}\ N)$ 都会相对比较小。
- 我们把 -1 也加进 B 里。

例子 2.4. 假设 $n=1829$ 以及 $\mathcal{B}=\{-1,2,3,5,7,11,13\}$. 计算 $\sqrt{n}=42.77,\sqrt{2n}=60.48,\sqrt{3n}=74.07,\sqrt{4n}=85.53$. 于是我们选取 $z=42,43,60,61,74,75,85,86$，分别计算 $z^2\bmod n$ over $\mathcal{B}$:

$$\begin{aligned} z_1{}^2 &\equiv 42^2\equiv -65\equiv(-1)\times 5\times 13\\ z_2^2 &\equiv 43^2\equiv 20\equiv 2^2\times 5\\ z_3{}^2 &\equiv 61^2\equiv 63\equiv 3^2\times 7\\ z_4{}^2 &\equiv 74^2\equiv -11\equiv(-1)\times 11\\ z_5{}^2 &\equiv 85^2\equiv -91\equiv(-1)\times 7\times 13\\ z_6^2 &\equiv 86^2\equiv 80\equiv 2^4\times 5. \end{aligned}$$

$$(42 \times 43 \times 61 \times 85)^2 \equiv (2 \times 3 \times 5 \times 7 \times 13)^2 (\mathrm{mod} 1829)$$

$$1459^2 \equiv 901^2 (\mathrm{mod} 1829).$$

$$\gcd(1459 + 901, 1829) = 59,$$

- Shor 算法, 是 Shor 提出的针对整数分解的高效量子算法 (在量子计算机上面运作的算法).

3 实验内容

- 1. **使用至少 3 种方法完成整数分解的程序（至少一种使用 NTL, 一种使用 Sage），并对附件中数据进行分解。**

- **2. 请将实现的代码和分解的结果写入实验报告。**

4 实验报告

完成实验报告，推荐学习使用 LaTex 。实验报告主要包含如下部分:
1. 报告题目，作者信息，每位组员完成的部分;
2. 报告摘要: 简述报告结果，例如可以做一个表格展示实验结果和时间;
3. 正文: 包括具体实验内容，使用的相关理论，结果的分析;
4. 参考文献。
最后将实验报告和代码打包提交。
11 月 11 号之前将实验报告 (命名格式: 组号 + 实验 4) 电子版发给助教,邮箱:202337022@mail.sdu.edu.cn