

密码学二级

分级通关系列教程

分解大整数

- 123456789011121314
- 12345678910111213141516171819202124
- 32799603723420868120030341899371
- 306116633122825106916047491910374116732637643679052793

分解大整数

■ 123456789011121314

```
N=123456789011121314
def findfactor(n):
    flag=0
    for i in range(2,int(RDF(sqrt(n)))+1):
        if n%i==0:
            flag=1
            findfactor(i)
            findfactor(n//i)
            break
    if flag==0:
        print(n)
        sys.stdout.flush
findfactor(N)
```

2

7

8818342072222951

大整数分解 p-1

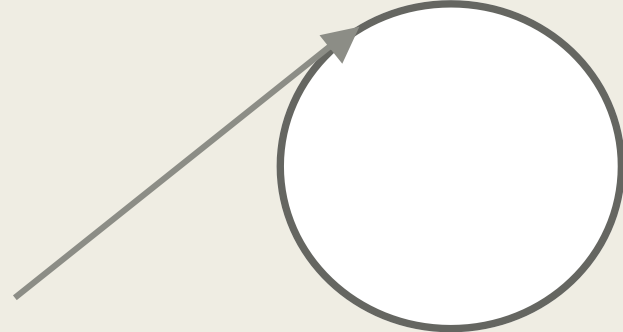
设 p 是奇素数, 那么 $2^{k(p-1)} \equiv 1 \pmod{p}$, 即 $p \mid 2^{k(p-1)} - 1$ 。如果 p 是 n 的一个素因子, 那么 $p \mid (2^{k(p-1)} - 1, n)$, 而 $(2^{k(p-1)} - 1, n) = ((2^{k(p-1)} - 1) \pmod{n}, n)$, 所以我们通过不断地求 $2^1 - 1, 2^{2!} - 1, \dots, 2^{B!} - 1 \pmod{p}$, 当 $p-1 \mid B!$ 的时候, 通过计算 $((2^{B!} - 1) \pmod{n}, n)$ 就可能求出 n 的非平凡因子。

- 当 $p-1 \mid B!$ 时, $p \mid 2^{p-1} - 1, 2^{p-1} - 1 \mid 2^{B!} - 1$

p-1

- `def factor_p_1(n,B):`
- `a=2`
- `for j in range(2,B+1):`
- `a=power_mod(a,j,n)`
- `d=gcd(a-1,n)`
- `if d>1:`
- `print(d)`
- `return`
- `factor_p_1(306116633122825106916047491910374116732637643679052793`
`,10**6)`

大整数分解Pollard- ρ



■ 基于生日攻击

假设 x_1, x_2, \dots, x_m 是 m 个整数形成的序列, p 是 n 的一个素因子, 如果序列中存在两个整数 x_i, x_j 使得 $p | x_i - x_j$, 那么, 通过计算 $(x_i - x_j, n)$ 就可能求出 n 的非平凡因子。而从 m 个整数中任取两个整数有 $\frac{m(m-1)}{2}$ 种取法, 当 m 较大时, 穷举计算变得较为困难。↵

Pollard ρ 方法首先选取任意 $x_1 = a < n$, 且计算 $x_{i+1} = x_i^2 + 1 \pmod n$, 如果将 x_1, x_2, \dots, x_m 看作一个随机序列, 根据生日悖论原理, 将近 $1.17\sqrt{p}$ 长的序列中存在 $x_i \equiv x_j \pmod p$ 的概率大于 50%。现在假设 x_i, x_j 是首次使得 $x_i \equiv x_j \pmod p$ 的两个整数, 且 $j > i$, 那么有 $x_{i+1} \equiv x_{j+1}, x_{i+2} \equiv x_{j+2}, \dots \pmod p$, 即序列从第 i 项开始, 以周期 $j - i$ 重复地关于 p 同余。设 $i \leq k < j$, 且 $j - i | k$ (连续 $j - i$ 个整数中必有一个是 $j - i$ 的倍数), 那么, 因为 k 是周期的倍数, 必然有 $x_k \equiv x_{2k} \pmod p$, 此时通过计算 $(x_k - x_{2k}, n)$ 就可能求出 n 的非平凡因子。↵

该方法为概率算法, 因为对于合数 n 而言, 其最小素因子 $p \leq n^{1/2}$, 所以算法的期望复杂度为 $O(n^{1/4})$ 。

Pollard- ρ

```
■ def factor_rho(n,a):  
■     b=(a*a+1)%n  
■     d=gcd(a-b,n)  
■     while d==1:  
■         a=(a*a+1)%n  
■         b=(b*b+1)%n  
■         b=(b*b+1)%n  
■         d=gcd(a-b,n)  
■     if d==n:  
■         print('fail')  
■     else:  
■         print(d)  
■ factor_rho(32799603723420868120030341899371,2)
```