密码学引论实验 5——因子分解

学号：202200460104

姓名：密语

班级：网安一班

2024 年 5 月 5 日

目录

1 任务一

1.1 任务描述

设 $p-q=2d>0$ 且 $n=pq$，试证明 $n+d^2$ 是完全平方数。

1.2 证明

由题意可知：

$$\begin{cases} p-q=2d>0 \\ n=pq \end{cases} \tag{1}$$

则有：

$$\begin{aligned} n+d^2 &= pq+d^2 \\ &= pq+\left(\frac{p-q}{2}\right)^2 \\ &= pq+\left(\frac{p^2-2pq+q^2}{4}\right) \\ &= \frac{4pq+p^2-2pq+q^2}{4} \\ &= \frac{p^2+2pq+q^2}{4} \\ &= \frac{(p+q)^2}{4} \end{aligned} \tag{2}$$

由于 $p-q=2d>0$，所以 $p>q$，则 $p+q>2q$，所以 $\frac{(p+q)^2}{4}$ 是完全平方数，即 $n+d^2$ 是完全平方数。

2 任务二

2.1 任务描述

设 $n=pq$ 是两个奇素数的乘积，给定小的正整数 d 满足 $n+d^2$ 是完全平方数。如何利用上述信息分解 n?

2.2 分解方法

由于 $n+d^2$ 是完全平方数，由上述证明可知：

$$n+d^2=\frac{(p+q)^2}{4} \tag{3}$$

可以遍历小整数 d，然后判断 $n+d^2$ 开平方后是否为整数，若是，则有：

$$\begin{cases} p+q=2\sqrt{n+d^2} \\ p-q=2d \end{cases} \tag{4}$$

解得：

$$\begin{cases} p=\sqrt{n+d^2}+d \\ q=\sqrt{n+d^2}-d \end{cases} \tag{5}$$

即可得到 n 的因子 p 和 q。

3 任务三

3.1 任务描述

利用上述方法分解 $n = 2189284635403183$。

3.2 分解过程

```python
import math


def is_square(num):
    sqrt_num = math.sqrt(num)
    if int(sqrt_num) ** 2 == num:
        return True
    else:
        return False


def factor(n):
    d = 0
    while True:
        if is_square((n + d * d) * 4):
            a = math.sqrt((n + d * d) * 4)
            b = 2 * d
            p = int((a + b) // 2)
            q = int((a - b) // 2)
            break

        d += 1
    return p, q


n = 2189284635403183
p, q = factor(n)
print("p =", p)
print("q =", q)
```

3.3 分解结果

运行上述代码，得到

```
D:\python\venv\Scripts\python.exe D:\python\RSA_test.py
p = 46789801
q = 46789783
```

图 1: 输出界面

$n = 2189284635403183$ 的因子为 $p = 46789801$ 和 $q = 46789783$。

4 任务四

4.1 任务描述

以下为 12 个合数，长度涵盖 30bit，40bit，50bit，请从 3 种长度的合数中各选择一个合数进行因子分解。要求：给出分解后的因子（16 进制），并提供 CPU、时间等关键测试数据。

4.2 系统 CPU 信息

图 2: CPU 信息

4.3 30bit 合数

选择 n=0x246d5fd1 进行分解，使用上面的方法即可进行分解。

```python
import math
import time


def is_square(num):
    sqrt_num = math.sqrt(num)
    if int(sqrt_num) ** 2 == num:
        return True
    else:
        return False


def factor(n):
    d = 0
    while True:
        if is_square((n + d * d) * 4):
            a = math.sqrt((n + d * d) * 4)
            b = 2 * d
            p = hex(int((a + b) // 2))
            q = hex(int((a - b) // 2))
            break

        d += 1
    return p, q


n = 0x2fbf76ea9f0b3
start_time = time.time()
p, q = factor(n)
end_time = time.time()
print("p =", p)
print("q =", q)
print("time:", end_time - start_time)
```

分解结果：

```
D:\python\venv\Scripts\python.exe D:\python\RSA_test.py
p = 0x7211
q = 0x51c1
time: 0.002058267593383789
```

图 3: 30bit 合数分解结果

4.4 40bit 合数

仍然采用上述的方法进行分解，选择 n=0x86a1755c41 进行分解。分解结果：

```
D:\python\venv\Scripts\python.exe D:\python\RSA_test.py
p = 0xd06bd
q = 0xa55d5
time: 0.04917263984680176
```

图 4: 40bit 合数分解结果

4.5 50bit 合数

仍然采用上述的方法进行分解，选择 n=0x2fbf76ea9f0b3 进行分解。分解结果：

```
D:\python\venv\Scripts\python.exe D:\python\RSA_test.py
p = 0x1e48d85
q = 0x1939ed7
time: 1.294424057006836
```

图 5: 50bit 合数分解结果