大数素性检测的数学理论基础分析

崔航

班级:2022211805 学号:2022211576

摘要

对于素数的研究一直是令数学家们着迷，如何对大素数进行准确地检测也是一直以来是密码学的重要研究方向。本文介绍了几种素数检测方法，分别是Fermat素数检测，Solovay-Strassen素数检测和Miller-Rabin素数检测等。并对这几种方法进行了代码实现。

关键词: 素数检测，伪素数，强伪素数

目录

1 Fermat小定理

当p为素数时，对于任意整数a，有$a^p \equiv a \pmod p$。即$a^{p-1} \equiv 1 \pmod p$。

素数检测的基石定理，十分重要。

2 伪素数

在检测素数时可能检测到符合素数特性的合数，在素数检测时需要格外注意。

2.1 Carmichael伪素数

由Fermat's Little Theorem可知，若对于任意整数a，有$a^p \equiv a \pmod p$，则p可能为素数。但是，若p为合数，仍有可能满足$a^p \equiv a \pmod p$。此时，p被称为Carmichael伪素数。

2.2 Fermat素数检测

1. 给定奇整数$n \geq 3$和安全参数t。
2. 随机选取整数b,$(b, n) = 1, 2 \leq b \leq n - 2$.
3. 计算$r = b^{n-1} \pmod{n}$。
4. 若$r \neq 1$，则n为合数。
5. 若$r = 1$，则n可能为素数，重复步骤2-4，共重复t次。

2.2.1 算法实现

```
def fermat(n, k):
    if n == 2:
        return True
    if n % 2 == 0:
        return False
    for i in range(k):
        a = random.randint(2, n - 2)
        if pow(a, n - 1, n) != 1:
            return False
    return True
```

2.3 平方因子检测

判断一个数是否为Carmichael伪素数，只需要判断其是否有平方因子。设n为一个有平方因子的整数，则存在整数a，$(a, n) = 1$，$a^{(n-1)} \not\equiv 1 \pmod{n}$。

2.3.1 算法实现

```
def square_factor(n):
    for i in range(2, int(n ** 0.5) + 1):
        if n % (i ** 2) == 0:
            return True
    return False
```

2.4 Euler伪素数

设n是一个正奇合数，设整数b与n互素，若$b^{\frac{n-1}{2}} \equiv (\frac{b}{n}) \pmod{n}$，其中$(\frac{b}{n})$为Jacobi符号，且$(\frac{b}{n}) \neq b^{\frac{n-1}{2}} \pmod{n}$，则$n$被称为Euler伪素数。则$n$被称为Euler伪素数。

2.4.1 Solovary-Strassen素数检测

1. 给定奇整数$n \geq 3$和安全参数t。
2. 随机选取整数b,$(b, n) = 1, 2 \leq b \leq n - 2$.
3. 计算$r = b^{\frac{n-1}{2}} \pmod{n}$。
4. 若$r \not\equiv \pm 1 \pmod{n}$，则$n$为合数。
5. 计算Jacobi符号$(\frac{b}{n})$。
6. 如果$r \neq s$，则n为合数。
7. 若$r \equiv \pm 1 \pmod{n}$，则n可能为素数，重复步骤2-4，共重复t次。

2.4.2 算法实现

```
def solovay_strassen(n, k):
    if n == 2:
        return True
    if n % 2 == 0:
        return False
    for i in range(k):
        a = random.randint(2, n - 2)
        if pow(a, (n - 1) // 2, n) != jacobi(a, n) % n:
            return False
    return True
```

3 强伪素数

设一个奇合数n,且有表达式$n-1=2^s d$，其中d为奇数。若对于任意整数a，有

$$a^d \equiv 1 \pmod n$$

或

$$a^{2^r d} \equiv -1 \pmod n$$

，则n被称为强伪素数。

3.1 Miller-Rabin素数检测

1. 给定奇整数$n \geq 3$和安全参数k。
2. d为奇整数，$n-1=2^s d$。
3. 随机选取整数b,$(b,n)=1, 2 \leq b \leq n-2$.
4. 计算$r \equiv b^d \pmod n$。
5. 若$r=1$或$r=n-1$，则n可能为素数，重复步骤2-4，共重复k次。
6. 否则$r_1 \equiv r^2 \pmod n$，若$r_1=n-1$，则n可能为素数，重复步骤2-4，共重复k次。
7. 否则计算$r_2 \equiv r_1^2 \pmod n$，若$r_2=n-1$，则n可能为素数，重复步骤2-4，共重复k次。
8. 依此类推，直到$r_{t-1}=n-1$，则n可能为素数，重复步骤2-4，共重复k次。
9. 若$r_{t-1} \neq n-1$，则n为合数。

3.1.1 算法实现

```
def miller_rabin(n, k):
    if n == 2:
        return True
    if n % 2 == 0:
        return False
    s = 0
    d = n - 1
    while d % 2 == 0:
        s += 1
        d //= 2
    for i in range(k):
        a = random.randint(2, n - 2)
```

```
        x = pow(a, d, n)
        if x == 1 or x == n - 1:
            continue
        for j in range(s - 1):
            x = pow(x, 2, n)
            if x == n - 1:
                break
        else:
            return False
    return True
```

4 总结

本文介绍了几种素数检测方法，分别是Fermat素数检测，Solovay-Strassen素数检测和Miller-Rabin素数检测。并对这几种方法进行了代码实现.除了这些实用的素数概率性检测，还有一些更加复杂的素数检测方法，如AKS素数检测，Lenstra素数检测等，安全性和实用性不高，本文不再赘述。

参考文献

[1] 刘学军,邢玲玲,林和平,粟浩然.Miller-Rabin素数检测优化算法研究与实现[J].信息技术,2008,32(12):141-143+147.

[2] 魏成行. 素性检测算法研究及其在现代密码学中的应用[D].山东大学,2010.

[3] 彭韬,陈文庆.基于VB的大素数Solovay-Strassen检测的设计与实现[J].电子技术与软件工程,2020,(10):161-162.

[4] 陈恭亮.信息安全数学基础[M].清华大学出版社,2004.