Date:11.11.2022

## Final Year B. Tech., Sem VII 2022-23

# **Cryptography And Network Security Lab**

# **Assignment submission**

PRN No: 2019BTECS00077

Full name: Avinash Vishnu Biradar

Batch: B7

**Assignment: 9** 

# Title of assignment: Implementation of Prime Factorization of Semiprime Number

#### Title:

Implementation of Prime Factorization of Semiprime Number

#### Aim:

To develop and implement Prime Factorization of Semiprime Number

## Theory:

- In mathematics, a semiprime is a natural number that is the product of exactly two prime numbers.
- The two primes in the product may equal each other, so the semiprimes include the squares of prime numbers. Because there are infinitely many prime numbers, there are also infinitely many semiprimes.
- Semiprimes are also called biprimes.

- Semiprimes that are not square numbers are called discrete, distinct, or squarefree semiprimes.
- The semiprimes less than 100 are:
  4, 6, 9, 10, 14, 15, 21, 22, 25, 26, 33, 34, 35, 38, 39, 46, 49, 51, 55, 57, 58, 62, 65, 69, 74, 77, 82, 85, 86, 87, 91, 93, 94, and 95
- A semiprime counting formula was discovered by E. Noel and G. Panos in 2005. Let Pi<sub>2</sub>(n) denote the number of semiprimes less than or equal to n. Then

$$\pi_2(n) = \sum_{k=1}^{\pi(\sqrt{n})} [\pi(n/p_k) - k + 1]$$

where Pi(x) is the prime-counting function and  $p_k$  denotes the kth prime.

```
Enter the Number:

3595
**

5
719

Process exited after 106.9 seconds with return value 0
Press any key to continue . . .
```

## **Python Code:**

```
import math
import time

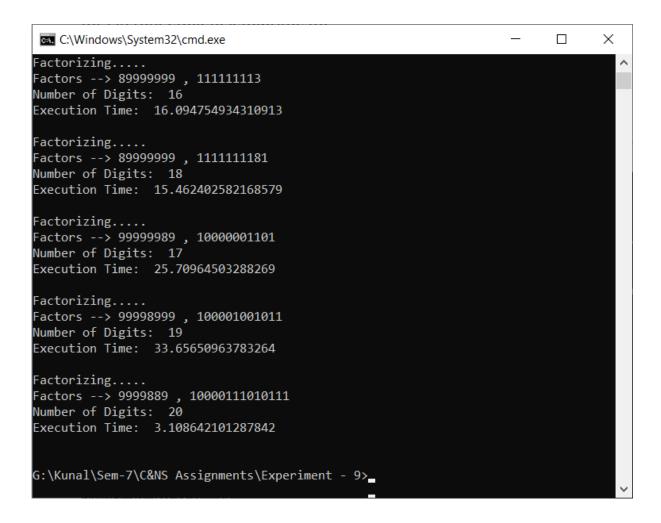
def getFactors(A,n):
    print("Factorizing.....")
    end = (int)(math.sqrt(A)) + 1;
    start_time = time.time()
    for i in range(2, end):
        if A % i == 0:
```

```
print(f"Factors --> {i} , {A // i}")
    break
end_time = time.time()
print("Number of Digits: ", n)
print("Execution Time: ", end_time - start_time)
print("")

semiprimes = [1000010666663, 10000079888899, 100000008888889,
1000000518888883, 100000000058888887, 100000005178888819,
1000000000099987889, 10000000000097987989, 100000000088787877679]
digits = [12, 13, 14, 15, 16, 18, 17, 19, 20]

for i in range(0,len(semiprimes)):
    getFactors(semiprimes[i],digits[i])
```

**Output:** 



#### **Conclusion:**

Performed the experiment successfully.

The program done above is for limited numbers till range of 10000 but there are codes which can factorize the numbers having 20-25 digits of range.