Course Name:	Applied Cryptography	Semester:	V
Date of Performance:	19 / 08 / 2025	DIV/ Batch No:	AC2
Student Name:	Aaryan Sharma	Roll No:	16010123012

Experiment No: 2

Title: Cryptographic Arithmetic

Aim and Objective of the Experiment:

To learn about Cryptographic Arithmetic and its implementation

1. Prime number generation
2. Primality testing
3. Prime Factorization

COs to be achieved:

CO2: Demonstrate and implement various Cryptographic Algorithms for securing systems

Books/ Journals/ Websites referred:

1. Stallings, W., Cryptography and Network Security: Principles and Practice, Second edition, Person Education
2. "Caesar Cipher in cryptography", https://www.geeksforgeeks.org/caesar-cipher-in-cryptography/, last retrieved on Aug 01, 2023
3. "PlayFair Cipher in cryptography": https://www.geeksforgeeks.org/playfair-cipher-with-examples/, last retrieved on Aug 01, 2023
4. "Transposition cipher in cryptology", https://www.britannica.com/topic/transposition-cipher, last retrieved on Aug 01, 2023

Theory:

Abstract:

Cryptographic arithmetic involves using mathematical operations and techniques within cryptography to secure communication and data. It includes concepts like modular arithmetic, one-way functions, prime numbers, exponentiation, and elliptic curve cryptography, all of which are crucial for ensuring the confidentiality, integrity, and authenticity of data in cryptographic systems.

Related Theory:

1. Number Theory: Number theory forms the foundation of many cryptographic algorithms. Concepts such as prime numbers, modular arithmetic, greatest common divisors, and Euler's totient function are central to understanding and implementing cryptographic arithmetic.

2. Modular Arithmetic: Modular arithmetic is fundamental in cryptographic operations. It involves performing arithmetic operations within a finite set, called a modulus. Theorems related to modular arithmetic, like Euler's Totient Theorem, are critical for cryptographic algorithms like RSA and Diffie-Hellman.

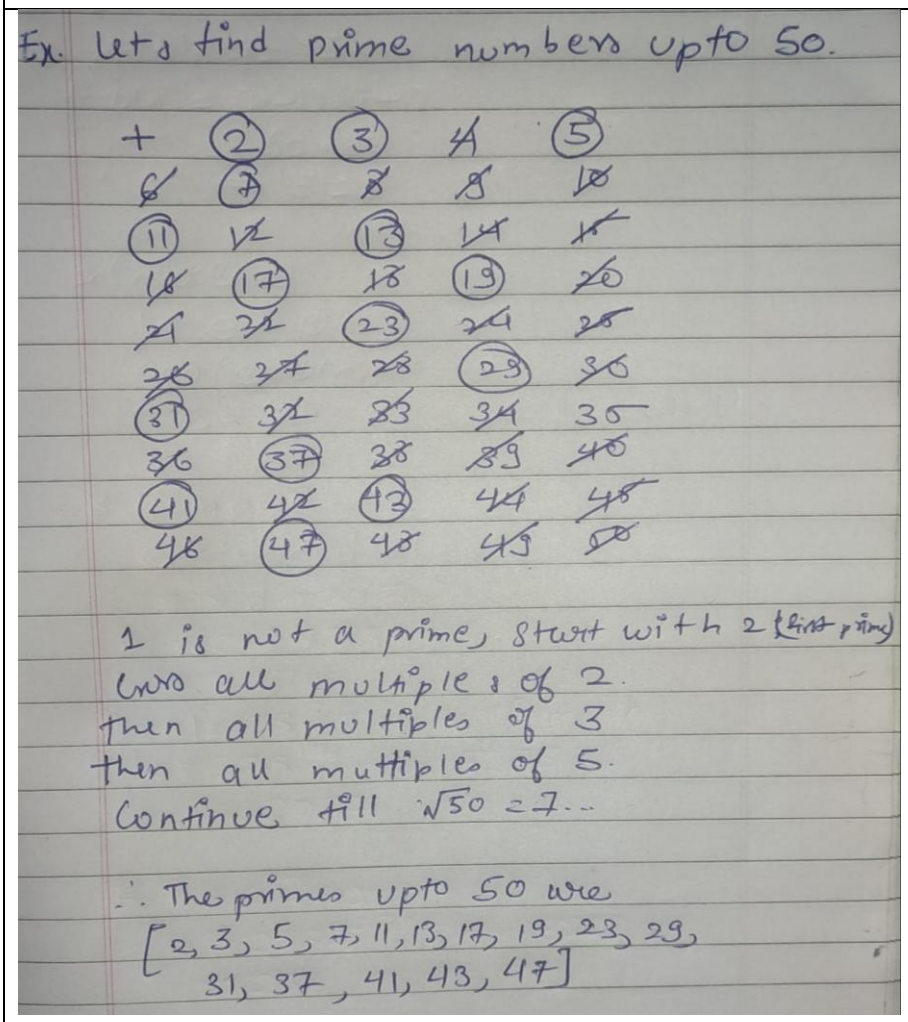
3. Additive Inverse: Two numbers a and b are additive inverses of each other if $a + b \equiv 0 \pmod{n}$. The additive inverse of a can be calculated as $b = n - a$.

4. Multiplicative Inverse: Two numbers a and b are the multiplicative inverse of each other if $a \times b \equiv 1 \pmod{n}$

Algorithm (Write assigned algo):

1. Create a list of integers from 2 to n.
2. Start with the first prime number, 2.
3. Eliminate all multiples of 2 (except 2 itself).
4. Move to the next non-crossed number (which will be a prime) and eliminate its multiples.
5. Repeat until all multiples up to root n are eliminated.
6. The remaining numbers in the list are primes.

Solve a small numerical for assigned algorithm (Paste photograph of same):



Code:

```python
def sieve_of_eratosthenes(n):
  primes = [True] * (n + 1)
  primes[0], primes[1] = False, False
  p = 2

  while p * p <= n:
    if primes[p]:
      for i in range(p * p, n + 1, p):
        primes[i] = False
    p += 1

  answer = []
  for i in range(n + 1):
    if primes[i]:
      answer.append(i)

  return answer

print("Prime numbers up to 1000:", sieve_of_eratosthenes(1000))
```

Output:

```
PS D:\KJSCE\BTech\TY\Sem V\AC\code> python Sieve_of_Eratosthenes.py
Prime numbers up to 1000: [2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53
, 59, 61, 67, 71, 73, 79, 83, 89, 97, 101, 103, 107, 109, 113, 127, 131, 137, 139, 14
9, 151, 157, 163, 167, 173, 179, 181, 191, 193, 197, 199, 211, 223, 227, 229, 233, 23
9, 241, 251, 257, 263, 269, 271, 277, 281, 283, 293, 307, 311, 313, 317, 331, 337, 34
7, 349, 353, 359, 367, 373, 379, 383, 389, 397, 401, 409, 419, 421, 431, 433, 439, 44
3, 449, 457, 461, 463, 467, 479, 487, 491, 499, 503, 509, 521, 523, 541, 547, 557, 56
3, 569, 571, 577, 587, 593, 599, 601, 607, 613, 617, 619, 631, 641, 643, 647, 653, 65
9, 661, 673, 677, 683, 691, 701, 709, 719, 727, 733, 739, 743, 751, 757, 761, 769, 77
3, 787, 797, 809, 811, 821, 823, 827, 829, 839, 853, 857, 859, 863, 877, 881, 883, 88
7, 907, 911, 919, 929, 937, 941, 947, 953, 967, 971, 977, 983, 991, 997]
```

Post Lab Subjective/Objective type Questions:

1. **Why cryptography is strongly based on prime numbers and modulo arithmetic?** Because problems like integer factorization and discrete logarithms (over primes and modular arithmetic) are mathematically hard, making them ideal for security. Modular arithmetic provides a finite cyclic structure for encryption, decryption, and key exchange.

2. **Discuss significance of primality testing algorithms.**
   They are used to quickly identify large prime numbers needed in RSA, Diffie-Hellman, etc. Efficient algorithms like Miller-Rabin and AKS make secure key generation practical and reliable.

3. **Compare and contrast prime number generation algorithms**
   - Trial Division: Simple but too slow for large primes.
   - Sieve methods: Efficient for small primes, not suitable for cryptographic primes.
   - Randomized + Primality Testing (Miller-Rabin): Fast, practical, widely used.
   - Deterministic (AKS): Guaranteed correctness but slower in practice.

Conclusion:

I have successfully completed the experiment on cryptographic arithmetic and prime number generation using the Sieve of Eratosthenes. Through this experiment, I learned how prime numbers form the foundation of cryptographic algorithms and how the sieve efficiently generates primes by eliminating multiples of smaller numbers.