Title: ShiftPrime-Cipher

A Lightweight Prime-Based Encryption Algorithm

Course Title: Mathematical Analysis for Computer Science
Course Code: CSE 361

Submitted By

MD. Tanjimul Haque Meet

Student ID: 2002049

Level: 3, Semester: II

Submitted To

Pankaj Bhowmik

Lecturer

Department of Computer Science & Engineering, HSTU

Hajee Mohammad Danesh Science and Technology University, Dinajpur-5200

1. Algorithm Design

Encryption Algorithm

This symmetric algorithm maps each digit of a numeric key to a unique prime number, which is used to shift the ASCII values of characters in the plaintext.

Prime Map

Digit	Prime
0	2
1	3
2	5
3	7
4	11
5	13
6	17
7	19
8	23
9	29

Decryption Algorithm

The decryption algorithm performs the inverse of the encryption by subtracting the prime-shifted value instead of adding it.

2. Pseudocode

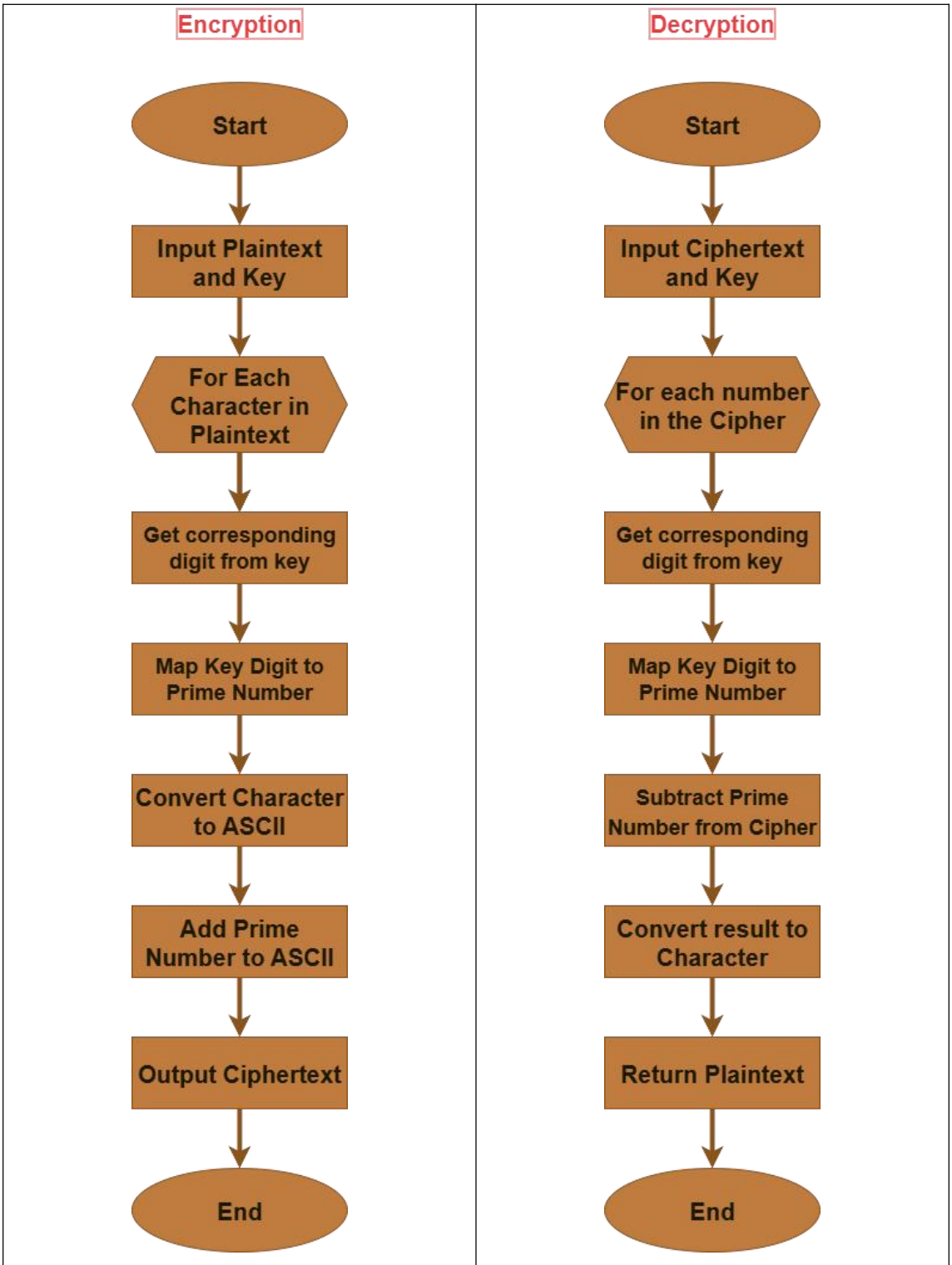
Encryption

1. Initialize an empty list called cipher.
2. For each character in the plaintext:
    - Get the corresponding digit from the key (cycled using modulo).
    - Map this digit to a prime number.
    - Convert the character to its ASCII value and add the prime number.
    - Append the result to the cipher list.
3. Return the cipher list.

Decryption

1. Initialize an empty string called plain.
2. For each number in the cipher:
    - Get the corresponding digit from the key (cycled using modulo).
    - Map this digit to a prime number.
    - Subtract the prime number from the encrypted value.
    - Convert the result to a character and append it to plain.
3. Return the plain text.

3. Flowcharts



Encryption Flowchart	Decryption Flowchart

4. Test Case:

Plaintext: HSTU CSE
Key: 4391

Encryption Process:

Index	Char	ASCII	Key	Prime Number	Encrypted Number
0	H	72	4	11	83
1	S	83	3	7	90
2	T	84	9	29	113
3	U	85	1	3	88
4		32	4	11	43
5	C	67	3	7	74
6	S	83	9	29	112
7	E	69	1	3	72

Ciphertext (as numbers): **83 90 113 88 43 74 112 72**

Decryption Process:

Index	Encrypted Number	Key	Prime Number	Decrypted ASCII	Decrypted Text
0	83	4	11	72	H
1	90	3	7	83	S
2	113	9	29	84	T
3	88	1	3	85	U
4	43	4	11	32	(space)
5	74	3	7	67	C
6	112	9	29	83	S
7	72	1	3	69	E

Decrypted Text: HSTU CSE

5. Source Code (Python)

```
def get_prime_shift(digit_char):
    prime_map = [2, 3, 5, 7, 11, 13, 17, 19, 23, 29]
    if not digit_char.isdigit():
        raise ValueError("Key must contain only digits.")
    return prime_map[int(digit_char)]

def encrypt(plaintext, key):
    cipher = []
    for i, ch in enumerate(plaintext):
        k = get_prime_shift(key[i % len(key)])
        cipher.append(ord(ch) + k)
    return cipher

def decrypt(cipher, key):
    plain = ''
    for i, val in enumerate(cipher):
        k = get_prime_shift(key[i % len(key)])
        plain += chr(val - k)
    return plain

def main():
    print("=== Prime-Based Cipher ===")
    choice = input("Do you want to (E) Encrypt or (D) Decrypt?
").strip().upper()

    if choice == 'E':
        text = input("Enter the plaintext: ")
        key = input("Enter the numeric key (e.g. 4391): ")
        if not key.isdigit():
            print("Error: Key must contain only digits.")
            return
        cipher = encrypt(text, key)
        print("Ciphertext (as numbers):", ' '.join(map(str,
cipher)))

    elif choice == 'D':
        cipher_input = input("Enter the cipher numbers separated
by spaces: ")
        try:
            cipher = list(map(int, cipher_input.strip().split()))
        except ValueError:
            print("Error: Cipher must be numbers separated by
spaces.")
            return
        key = input("Enter the numeric key (e.g. 4391): ")
        if not key.isdigit():
            print("Error: Key must contain only digits.")
            return
        plain = decrypt(cipher, key)
        print("Decrypted Text:", plain)
    else:
        print("Invalid choice. Please select E or D.")

if __name__ == "__main__":
    main()
```

6. Conclusion

The **ShiftPrime-Cipher** presents a lightweight and effective encryption technique grounded in elementary number theory. By leveraging prime numbers for character shifting, it introduces irregularity that strengthens the cipher against simple pattern-based attacks. This algorithm is particularly suitable for educational purposes and lightweight applications where simplicity and clarity are prioritized over industrial-grade security.

7. References

- Burton, D. M. *Elementary Number Theory*. McGraw-Hill Education.
- Stallings, W. *Cryptography and Network Security: Principles and Practice*. Pearson.
- GeeksforGeeks. (n.d.). *Prime Numbers – Basics and Applications*. https://www.geeksforgeeks.org/prime-numbers/
- Tutorialspoint. (n.d.). *Cryptography Basics*. https://www.tutorialspoint.com/cryptography/index.htm
- draw.io. (n.d.). *Free Online Diagram Software*. https://www.draw.io/