CNS ASSIGNMENT 4 - IMPLEMENTATION OF OWN CRYPTOGRAPHIC ALGORITHM : CRYPTIT

MIS - 112003151

NAME - ADWAIT VIPRA

The `cryptit` program defines a simple encryption and decryption scheme using a key. Here's a brief description of each function and the main program:

1. `xenc(ch, key)` : XOR-based encryption function that takes a character `ch` and a key, performs bitwise XOR, and adds the key to the result.
2. `xdec(cd, key)` : XOR-based decryption function that takes a ciphered character `cd` and a key, subtracts the key, and then performs bitwise XOR.
3. `encrypt(msg, key)` : Takes a message `msg` and a key, iterates through each character in the message, encrypts it using the `xenc` function, and constructs the ciphertext.
4. `decrypt(cph, key)` : Takes a ciphertext `cph` and a key, iterates through each character in the ciphertext, decrypts it using the `xdec` function, and constructs the decrypted message.
5. `main()` : The main program where the user is prompted to enter a message and a key. It then encrypts and decrypts the message using the provided key and prints the original message, ciphertext, and the decrypted message.

In [1]:
```python
#!/bin/python

def xenc(ch, key):
    cd = (key ^ ch) + key
    return cd

def xdec(cd, key):
    ch = (cd - key) ^ key
    return ch

def encrypt(msg, key):
    cph = ""
    for ch in msg:
        cph += chr(xenc(ord(ch), key) % 0x110000)
    return cph

def decrypt(cph, key):
    ans = ""
    for cd in cph:
        ans += chr(xdec(ord(cd), key) % 0x110000)
    return ans

def main():
    msg = input("Enter Message: ")
    key = int(input("Enter Key (Integer): "))

    cph = encrypt(msg, key)
    ans = decrypt(cph, key)

    print("Encrypted Plaintext:", msg)
    print("Ciphertext:", cph)
    print("Decrypted Plaintext:", ans)

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

```
Enter Message: SECRET
Enter Key (Integer): 1729
Encrypted Plaintext: SECRET
Ciphertext: ൓൅ൃൔ൅ൖ
Decrypted Plaintext: SECRET
```