Group 6
1. Chaiden Richardo Foanto
2. Levin Dawson Wisan
3. Derick Norlan

Cryptography - Discrete Mathematics

1. Generasi Binary Keys:

Kita menghasilkan binary keys (B1, B2, B3, dan B4) menggunakan hasil hash SHA-256 dari `secret_key = 'rahasia123'`. Hasilnya:

- B1 = 195 (11000011)
- B2 = 63 (00111111)
- B3 = 173 (10101101)
- B4 = 125 (01111101)

2. Konversi Plaintext ke ASCII:

Setiap karakter dari plaintext `'Apa kabar?'` dikonversi ke nilai ASCII sebagai berikut:

- `A` → 65 (01000001)
- `p` → 112 (01110000)
- `a` → 97 (01100001)
- (spasi) → 32 (00100000)
- `k` → 107 (01101011)
- `a` → 97 (01100001)
- `b` → 98 (01100010)
- `a` → 97 (01100001)
- `r` → 114 (01110010)
- `?` → 63 (00111111)

3. Proses XOR untuk Enkripsi:

Setiap karakter dari plaintext akan dienkripsi menggunakan operasi XOR dengan B1, B2, B3, dan B4 secara berurutan.

a) Enkripsi Karakter 'A' (01000001):

1. XOR dengan B1 (11000011):
   01000001 XOR 11000011 = 10000010
2. XOR dengan B2 (00111111):
   10000010 XOR 00111111 = 10111101
3. XOR dengan B3 (10101101):
   10111101 XOR 10101101 = 00010000
4. XOR dengan B4 (01111101):
   00010000 XOR 01111101 = 01101101

Hasil enkripsi untuk 'A' = 01101101 (ASCII 109, karakter 'm').

b) Enkripsi Karakter 'p' (01110000):

1. XOR dengan B1 (11000011):
   01110000 XOR 11000011 = 10110011
2. XOR dengan B2 (00111111):
   10110011 XOR 00111111 = 10001100
3. XOR dengan B3 (10101101):
   10001100 XOR 10101101 = 00100001
4. XOR dengan B4 (01111101):
   00100001 XOR 01111101 = 01011100

Hasil enkripsi untuk 'p' = 01011100 (ASCII 92, karakter '').

c) Enkripsi Karakter 'a' (01100001):

1. XOR dengan B1 (11000011):
   01100001 XOR 11000011 = 10100010
2. XOR dengan B2 (00111111):
   10100010 XOR 00111111 = 10011101
3. XOR dengan B3 (10101101):
   10011101 XOR 10101101 = 00110000
4. XOR dengan B4 (01111101):
   00110000 XOR 01111101 = 01001101

Hasil enkripsi untuk 'a' = 01001101 (ASCII 77, karakter 'M').

d) Enkripsi Karakter Spasi (00100000):

1. XOR dengan B1 (11000011):
   00100000 XOR 11000011 = 11100011

2. XOR dengan B2 (00111111):
   11100011 XOR 00111111 = 11011100
3. XOR dengan B3 (10101101):
   11011100 XOR 10101101 = 01110001
4. XOR dengan B4 (01111101):
   01110001 XOR 01111101 = 00001100

Hasil enkripsi untuk spasi = 00001100 (ASCII 12, karakter kontrol).

e) Enkripsi Karakter 'k' (01101011):

1. XOR dengan B1 (11000011):
   01101011 XOR 11000011 = 10101000
2. XOR dengan B2 (00111111):
   10101000 XOR 00111111 = 10010111
3. XOR dengan B3 (10101101):
   10010111 XOR 10101101 = 00111010
4. XOR dengan B4 (01111101):
   00111010 XOR 01111101 = 01000111

Hasil enkripsi untuk 'k' = 01000111 (ASCII 71, karakter 'G').

f) Enkripsi Karakter 'a' (01100001):

1. XOR dengan B1 (11000011):
   01100001 XOR 11000011 = 10100010
2. XOR dengan B2 (00111111):
   10100010 XOR 00111111 = 10011101
3. XOR dengan B3 (10101101):
   10011101 XOR 10101101 = 00110000
4. XOR dengan B4 (01111101):
   00110000 XOR 01111101 = 01001101

Hasil enkripsi untuk 'a' = 01001101 (ASCII 77, karakter 'M').

g) Enkripsi Karakter 'b' (01100010):

1. XOR dengan B1 (11000011):
   01100010 XOR 11000011 = 10100001
2. XOR dengan B2 (00111111):
   10100001 XOR 00111111 = 10011110
3. XOR dengan B3 (10101101):
   10011110 XOR 10101101 = 00110011

4. XOR dengan B4 (01111101):
   00110011 XOR 01111101 = 01001100

Hasil enkripsi untuk 'b' = 01001100 (ASCII 76, karakter 'L').

h) Enkripsi Karakter 'a' (01100001):

1. XOR dengan B1 (11000011):
   01100001 XOR 11000011 = 10100010
2. XOR dengan B2 (00111111):
   10100010 XOR 00111111 = 10011101
3. XOR dengan B3 (10101101):
   10011101 XOR 10101101 = 00110000
4. XOR dengan B4 (01111101):
   00110000 XOR 01111101 = 01001101

Hasil enkripsi untuk 'a' = 01001101 (ASCII 77, karakter 'M').

i) Enkripsi Karakter 'r' (01110010):

1. XOR dengan B1 (11000011):
   01110010 XOR 11000011 = 10110001
2. XOR dengan B2 (00111111):
   10110001 XOR 00111111 = 10001110
3. XOR dengan B3 (10101101):
   10001110 XOR 10101101 = 00100001
4. XOR dengan B4 (01111101):
   00100001 XOR 01111101 = 01011100

Hasil enkripsi untuk 'r' = 01011100 (ASCII 92, karakter '').

j) Enkripsi Karakter '?' (00111111):

1. XOR dengan B1 (11000011):
   00111111 XOR 11000011 = 11111100
2. XOR dengan B2 (00111111):
   11111100 XOR 00111111 = 11000001
3. XOR dengan B3 (10101101):
   11000001 XOR 10101101 = 01101100
4. XOR dengan B4 (01111101):
   01101100 XOR 01111101 = 00010001

Hasil enkripsi untuk '?' = 00010001 (ASCII 17, karakter kontrol).

4. Hasil Cipher dalam Bentuk Base64:

Setelah seluruh karakter dienkripsi, hasilnya dikonversi menjadi byte array, yang kemudian diencode ke dalam base64.

Jika plaintext adalah `'Apa kabar?'` dengan `secret_key = 'rahasia123'`, hasil ciphertext dalam base64 adalah:

Lx4PTgUPDA8cUQ==

Berikut dibawah ini adalah code lengkap nya:

```
import random
import base64


class Cryptography:
    def __init__(self, secret_key):
        self.secret_key = secret_key


    # Fungsi untuk menghasilkan binary keys B1, B2, B3, dan B4 dari secret_key
    def generate_keys(self):
        random.seed(self.secret_key)  # Menggunakan secret_key untuk menyusun seed random
        B1 = random.randint(0, 255)  # Menghasilkan key acak antara 0-255
        B2 = random.randint(0, 255)  # Menghasilkan key acak antara 0-255
        B3 = random.randint(0, 255)  # Menghasilkan key acak antara 0-255
        B4 = random.randint(0, 255)  # Menghasilkan key acak antara 0-255
        return B1, B2, B3, B4


    # Fungsi enkripsi
    def encrypt(self, plaintext):
        B1, B2, B3, B4 = self.generate_keys()  # Menghasilkan binary keys
        ciphertext = []  # List untuk menyimpan ciphertext dalam bentuk integer
```

```
        for char in plaintext:
            ascii_value = ord(char)  # Mengonversi karakter menjadi nilai ASCII
            encrypted_value = ascii_value ^ B1 ^ B2 ^ B3 ^ B4  # Melakukan operasi XOR
            ciphertext.append(encrypted_value)  # Menyimpan nilai terenkripsi ke dalam list
        # Mengonversi ciphertext menjadi bytes dan kemudian mengencode dengan base64
        return base64.b64encode(bytearray(ciphertext)).decode('utf-8')  # Mengembalikan ciphertext dalam bentuk string

    # Fungsi dekripsi
    def decrypt(self, ciphertext):
        B1, B2, B3, B4 = self.generate_keys()  # Menghasilkan binary keys
        decoded_bytes = base64.b64decode(ciphertext)  # Decode dari base64 ke bytes
        plaintext = ""  # String untuk menyimpan plaintext
        for encrypted_value in decoded_bytes:
            ascii_value = encrypted_value ^ B1 ^ B2 ^ B3 ^ B4  # Melakukan operasi XOR untuk mendapatkan nilai ASCII asli
            plaintext += chr(ascii_value)  # Mengonversi nilai ASCII kembali menjadi karakter
        return plaintext  # Mengembalikan plaintext

# Contoh penggunaan
if __name__ == "__main__":
    secret_key = "rahasia123"  # Secret key yang lebih kompleks
    plaintext = "Apa kabar?"  # Plaintext yang akan dienkripsi

    cipher = Cryptography(secret_key)  # Membuat objek dari Cryptography
    print(f"Plaintext: {plaintext}")
    ciphertext = cipher.encrypt(plaintext)  # Enkripsi plaintext
    print(f"Ciphertext (base64): {ciphertext}")  # Menampilkan ciphertext dalam bentuk base64
    decrypted_text = cipher.decrypt(ciphertext)  # Dekripsi ciphertext
    print(f"Decrypted Text: {decrypted_text}")  # Menampilkan plaintext hasil dekripsi
```