Laporan Tugas 1 Kripto

Teori Singkat

a. Caesar Cipher

Caesar cipher adalah algoritma substitusi sederhana yang mengenkripsi pesan dengan menggeser setiap huruf pada plainteks sejumlah tetap (kunci) pada alfabet, misalnya geser 3 posisi ke kanan. Jika kunci 3, maka A menjadi D, B menjadi E, dan seterusnya. Dekripsi dilakukan dengan menggeser ke arah sebaliknya.

b. Viginere Cipher

Vigenère cipher adalah cipher substitusi polialfabetik yang menggunakan sebuah kata kunci untuk menentukan jumlah pergeseran setiap huruf. Setiap karakter pada plainteks digeser berdasarkan huruf kunci yang berulang sesuai panjang pesan, sehingga membuat pola lebih sulit ditebak dibanding Caesar.

c. Affine Cipher

Affine cipher memperluas Caesar cipher dengan menerapkan fungsi matematika $C \equiv mP+b \pmod{n}$ di mana P adalah huruf pada plainteks yang diubah dengan mengalikan oleh m dan menambahkan b, lalu hasilnya diambil modulo n (jumlah alfabet). Agar dapat didekripsi, m harus coprime dengan n.

d. Playfair Cipher

Playfair cipher mengenkripsi pasangan (digram) huruf dan menggunakan matriks 5x5 berisi alfabet yang diubah urutannya berdasarkan kunci. Teks dipecah jadi digram, lalu dipetakan menurut aturan posisi relatif dalam matriks, sehingga membuatnya lebih kuat dari substitusi monoalfabetik.

e. Hill Cipher

Hill cipher menggunakan prinsip aljabar linear dengan matriks. Setiap blok plainteks (misal dua atau tiga huruf) dikonversi menjadi vektor, lalu dikalikan dengan matriks kunci, hasilnya diambil modulo jumlah alfabet. Matriks kunci harus invertibel agar dekripsi memungkinkan.

Aplikasi Algoritma *Classic Cipher* (Input – Output)

Caesar Cipher

```python
def caesar_encrypt(text, shift):
    result = ''
    for char in text:
        if char.isalpha():
            base = ord('A') if char.isupper() else ord('a')
            result += chr((ord(char) - base + shift) % 26 + base)
        else:
            result += char
    return result

print(caesar_encrypt('KRIPTO', 4))
#OUTPUT : OVMTXS
```

Vigenere Cipher

```python
def vigenere_encrypt(plain, key):
    key = key.upper()
    result = ''
    for i, char in enumerate(plain.upper()):
        if char.isalpha():
            shift = ord(key[i % len(key)]) - 65
            result += chr((ord(char) - 65 + shift) % 26 + 65)
        else:
            result += char
    return result

print(vigenere_encrypt('HELLO', 'ADILLA'))
#OUTPUT : HHTWZ
```

Playfair Cipher

```python
def generate_table(key):
    alphabet = 'ABCDEFGHIKLMNOPQRSTUVWXYZ'
    table = ''
    for c in key.upper() + alphabet:
        if c not in table:
            table += c
    return [table[i:i + 5] for i in range(0, 25, 5)]

table = generate_table('KEYWORD')
for row in table:
    print(row)

#Ouput
"""
KEYWO
RDABC
FGHIL
MNPQS
TUVXZ
"""
```

Hill Cipher

```python
import numpy as np

def hill_encrypt(text, key):
    text = text.upper().replace(' ', '')
    n = int(len(key)**0.5)
    key = np.array(key).reshape(n, n)

    # Padding teks jika perlu
    if len(text) % n != 0:
        text += 'X' * (n - len(text) % n)

    result = ''
    for i in range(0, len(text), n):
        block = [ord(c) - 65 for c in text[i:i+n]]
        cipher = np.dot(key, block) % 26
        result += ''.join(chr(c + 65) for c in cipher)
    return result

print(hill_encrypt('TEST', [3, 3, 2, 5]))

#OUTPUT : RGHB
```

Affine Cipher

```python
def affine_encrypt(text, a, b):
    result = ''
    for char in text.upper():
        if char.isalpha():
            result += chr(((a * (ord(char) - 65) + b) % 26) + 65)
        else:
            result += char
    return result

print(affine_encrypt('HELLO', 5, 8))
#OUTPUT : RCLLA
```

Analisis Kelemahan

a. Caesar Cipher

- Mudah dipecahkan dengan brute-force karena jumlah kunci hanya 25 kemungkinan.
- Pola frekuensi huruf tidak berubah, sehingga gampang untuk frequency analysis.
- Tidak cocok untuk pesan panjang dan data sensitif, sangat lemah terhadap cryptanalysis.

b. Vigenere Cipher

- Rentan jika kunci terlalu pendek atau berulang (serangan Kasiski).
- Pola masih bisa dianalisis jika panjang kunci diketahui.
- Tidak cukup kuat untuk melindungi data modern, apalagi tanpa kunci yang benar-benar acak dan panjang.

c. Affine Cipher

- Kunci terbatas sehingga dapat dipecahkan dengan brute-force.
- Pola substitusi masih bisa dianalisis dengan frequency analysis.
- Jika sebagian plainteks dan ciphertext diketahui, kunci bisa dicari dengan persamaan matematika.

d. Playfair Cipher

- Pola digram (dua huruf) tetap dapat dianalisis dan dipecahkan.
- Penggunaan matriks menyebabkan ambiguitas jika huruf dihapus (misal huruf J).
- Ciphertext masih memiliki pola statistik mirip plaintext.

e. Hill Cipher

- Rentan terhadap known-plaintext attack jika beberapa blok diketahui.
- Jika matriks kunci tidak invertibel (mod 26), teks tidak bisa didekripsi.
- Blok plainteks yang sama, hasil chiffrenya selalu sama sehingga ada pola.

Kesimpulan :

Cipher klasik seperti Caesar, Vigenère, Affine, Playfair, dan Hill kini dianggap tidak aman karena mudah dipecahkan dengan teknik modern. Hill dan Playfair sedikit lebih kuat, tapi semua cipher klasik hanya cocok untuk edukasi, bukan perlindungan data modern seperti saat ini.

Link Github :
https://github.com/Adillare16/Kriptografi/blob/main/cipher_klasik.ipynb