Nama : Anna Adelia Dewanta
NIM : 222111904
Kelas : 3SI2

KEAMANAN SISTEM INFORMASI

Cryptography

Buat Algoritma, hitung *worse case* kompleksitas waktu (O big O) dari masing-masing algoritma, program (*encrpyt* dan *decrypt*) cipher.

Kode dapat diakses melalui tautan berikut.

https://github.com/annadewanta/Tugas-KSI-Pertemuan-5.git

1. Transposition Cipher

Key berupa angka

Algoritma Enkripsi:

1) Pengguna diminta untuk memasukkan pesan yang akan dienkripsi dan kunci enkripsi.
2) Program akan mengonversi pesan menjadi matriks dengan jumlah kolom yang sama dengan nilai kunci.
3) Setiap karakter pesan dimasukkan ke dalam matriks secara berurutan berdasarkan kolom, dimulai dari kolom pertama hingga terakhir.
4) Setelah matriks terisi, program akan membaca karakter dari matriks secara kolom-kolom dan menggabungkannya menjadi pesan terenkripsi.
5) Pesan terenkripsi kemudian ditampilkan kepada pengguna.

Penghitungan kompleksitas waktu:

- Iterasi untuk mengonversi pesan menjadi matriks berlangsung sebanyak len(message) kali.
- Iterasi untuk menggabungkan karakter matriks menjadi pesan terenkripsi berlangsung sebanyak key kali.

Jadi, kompleksitas waktu algoritma enkripsi adalah O(len(message) + key).

Algoritma Dekripsi:

1) Pengguna diminta untuk memasukkan pesan terenkripsi dan kunci dekripsi.
2) Program akan menghitung jumlah kolom yang dibutuhkan berdasarkan panjang pesan terenkripsi dan nilai kunci.
3) Pesan terenkripsi dipecah menjadi matriks dengan jumlah kolom yang sesuai dengan nilai kunci.

4) Setiap karakter pesan terenkripsi dimasukkan ke dalam matriks secara berurutan berdasarkan kolom, dimulai dari kolom pertama hingga terakhir.
5) Setelah matriks terbentuk, program membaca karakter dari matriks secara baris-baris dan menggabungkannya menjadi pesan terdekripsi.
6) Pesan terdekripsi kemudian ditampilkan kepada pengguna.

Penghitungan kompleksitas waktu:

- Iterasi untuk memecah pesan terenkripsi menjadi matriks: len(message).
- Iterasi untuk menggabungkan karakter matriks menjadi pesan terdekripsi: numOfColumns kali

Jadi, kompleksitas waktu algoritma dekripsi adalah O(len(message) + numOfColumns).

Key berupa kata tanpa huruf berulang

Algoritma Enkripsi:

1) Pesan yang dimasukkan diubah menjadi matriks dengan jumlah kolom yang sama dengan panjang kunci.
2) Matriks tersebut diisi dengan pesan dan karakter '_'.
3) Matriks dibaca secara kolom-kolom berdasarkan urutan alfabetik karakter kunci.
4) Karakter-karakter dari matriks dibaca dan digabungkan menjadi pesan terenkripsi.

Penghitungan kompleksitas waktu:

- Iterasi untuk mengonversi pesan menjadi matriks: O(len(message))
- Iterasi untuk menggabungkan karakter matriks menjadi pesan terenkripsi: O(len(message))

Jadi, kompleksitas waktu algoritma enkripsi adalah O(len(message))

Algoritma Dekripsi:

1) Pesan terenkripsi yang dimasukkan dipecah menjadi matriks dengan jumlah kolom yang sama dengan panjang kunci.
2) Matriks tersebut diisi dengan karakter-karakter dari pesan terenkripsi.
3) Matriks dibaca secara kolom-kolom berdasarkan urutan alfabetik karakter kunci.
4) Karakter-karakter dari matriks dibaca dan digabungkan menjadi pesan terdekripsi.

Penghitungan kompleksitas waktu:

- Iterasi untuk memecah pesan terenkripsi menjadi matriks: O(len(cipher))
- Iterasi untuk menggabungkan karakter matriks menjadi pesan terdekripsi: O(len(cipher))

Jadi, kompleksitas waktu algoritma dekripsi adalah O(len(cipher))

2. Substitution Cipher

Algoritma Enkripsi:

1) Buat sebuah dictionary (dict1) yang berisi setiap karakter dalam alfabet sebagai kunci dan karakter setelah digeser sebesar key sebagai nilai.
2) Iterasi melalui pesan teks yang akan dienkripsi.
3) Jika karakter adalah huruf, maka gantilah karakter tersebut dengan karakter yang sesuai dari dictionary dict1. Jika bukan huruf, biarkan karakter tersebut tidak berubah.
4) Gabungkan semua karakter yang telah dienkripsi menjadi sebuah string yang menjadi pesan terenkripsi.

Penghitungan kompleksitas waktu:

- Iterasi melalui pesan teks memerlukan waktu linear terhadap panjang pesan: (len(text))
- Menggabungkan karakter menjadi pesan terenkripsi memerlukan waktu linear terhadap panjang pesan: (len(text))

Jadi, kompleksitas waktu algoritma enkripsi adalah O(len(text))

Algoritma Dekripsi:

1) Buat sebuah dictionary (dict2) yang berisi setiap karakter dalam alfabet sebagai kunci dan karakter sebelum digeser sebesar key sebagai nilai.
2) Iterasi melalui pesan terenkripsi yang akan didekripsi.
3) Jika karakter adalah huruf, maka gantilah karakter tersebut dengan karakter yang sesuai dari dictionary dict2. Jika bukan huruf, biarkan karakter tersebut tidak berubah.
4) Gabungkan semua karakter yang telah didekripsi menjadi sebuah string yang menjadi pesan terdekripsi.

Penghitungan kompleksitas waktu:

- Iterasi melalui pesan terenkripsi memerlukan waktu linear terhadap panjang pesan terenkripsi: (len(cipher_text))
- Menggabungkan karakter menjadi pesan terdekripsi memerlukan waktu linear terhadap panjang pesan terenkripsi: (len(cipher_text))

Jadi, kompleksitas waktu algoritma dekripsi adalah O(len(cipher_text))