CSE350/550: Network Security

Programming Assignment no. 1 - Project 2

~ Ankit Kumar(2021015) & Lakshay Chauhan(2021060)

1. Introduction

We were required to develop executable programs to encrypt, decrypt, and launch a brute-force attack to discover the key given a cipher text. The encryption and decryption of the plaintext and ciphertext must be done using a transposition algorithm.

The whole code is essentially divided into 4 parts or functions.

- encrypt
- decrypt
- hash
- brute force

2. Plaintext, Key and Ciphertext

Plaintext

The plaintext is a sequeunce of english alphabets {a,b, ...,x, y} without spaces. The length of the plaintext is not limited to any number but it is advised to keep it small enough to apply the bruteforce in meaningful time. The plaintext should also be long enough to make it meaningful to excrypt and such that length of plaintext is greater than the key length.

Key

Key can be considered as a tuple, where 1st element is a integer: length of the key and 2nd element of of the key is list of numbers consisting the sequence of column in which the text should be read and concatinate. The reverse order of this sequence should be used to decrypt the text.

Ciphertext

The Chiphertext is the encrypted text returned by the decrypt function. It is will longer than both, the initial plaintext and the hash. The chipher text is also padded with the letter '**z**'.

3. Encryption

This code implements a transposition cipher using Assembly (encrypt.asm) and C (transposition_cipher.c). A transposition cipher is a cryptographic method where character positions in plaintext are shifted according to a systematic approach to form ciphertext.

- Assembly Code (encrypt.asm)
  The encrypt routine starts by saving registers and setting up local variables for the plaintext and key. The length of the plaintext and key are stored in `plaintext.len` and `key.len`. The plaintext is then aligned and padded with 'z' to match the length of the ciphertext.
  The key's length is used to calculate the length of the ciphertext(`ciphertext.len`). The transposition process involves rearranging the plaintext characters according to the key. Each character's position in the plaintext is shifted to a new position in the ciphertext. This rearrangement is based on the key values, ensuring each character from the plaintext is correctly positioned in the ciphertext. After the transposition, the ciphertext pointer `ciphertext.ptr` holds the final encrypted message.

- C Code (transposition.c)
  The main function in the C code checks if the operation is encrypt. It then processes the key and the plaintext, constructing an integer array for the key based on command-line arguments. A hash of the plaintext is concatenated with the original plaintext, and this modified plaintext is passed to the encrypt function along with its length and the key. The resulting ciphertext is printed, and any dynamically allocated memory is freed.

Usage

The usage for the binary files and the supported commands can be found in the documentation file present in the dir transposition_cipher