Assignment 2 - Data Security

Pedro Augusto Ennes de Martino Camargo - PG59791

Question 1

A forma que usei para descobrir a mensagem decifrada foi utilizar um script com o propósito de conseguir dados estatísticos e utilizá-los com as tabelas de frequência das letras, dígrafos e trígafos mais comuns na língua portuguesa.

```
permutation {
    "H": "a",
    "P": "o",
    "W": "r",
    "K": "p",
    "F": "f",
    "R": "n",
    "D": "e",
    "T": "m",
    "N": "u",
    "L": "q",
    "O": "l",
    "A": "v",
    "J": "d",
    "Y": "c",
    "S": "f",
    "V": "i",
    "Q": "s",
    "I": "b",
    "X": "h",
    "U": "x",
    "G": "z",
    "C": "g",
    "B": "j",
}
```

Com essa estratégia, eu consegui mapear todas as letras cifradas para a sua equivalente na mensagem decifrada.

Primeiramente, como na aula o professor nos deu que H → "a", e também pude ver que a letra "a" é a mais comum na mensagem, esse mapeamento foi instantâneo.

Depois, observei no meu script que aW/Wa eram muito comuns na mensagem, então fui ver na tabela de dígrafos e haviam dois dígrafos muito parecidos: ar/ra. Então, W → "r"

Com os mapeamentos, observei que havia a fraquência <Kar> 9 vezes na mensagem, e na tabela de trígafos havia o trígafo "par" comum na lingua portuguesa, o único que termina com "ar". Então K → "p".

Com o tempo, várias letras foram sendo eliminadas através das tabelas de frequência, e, consequentemente, consegui ler algumas palavras na mensagem cifrada. Como por exemplo: IrVrNTapPrtaDQtPNYDCP: pPrta → "porta". Então P → "o".

Utilizei a mesma estratégia sempre, até no final ter todas as letras mapeadas.

Chave → HIYJDSCXVBOTRPKLWQFNAUG

Resultado final:

odiscoamareloiluminousedoisdosautomoveisdafrenteaceleraramantesqueosinalvermelhoaparecessenapassadeiradepeonssurgiuodesenhodohomemverde
agentequeesperavacomecoutaatravessararuapisandoasfaixasbrancaspintadasnacapanegadoasfaltononahanadaquemenosseparecacomumazebrapore
masimlhechamamosautomobilistasimpacientescomopenopedaldaembraiagemmantinhamemtensaooscarrosavancandorecuandocomocavalosnervososq
uesentissemvirnoarachibataospeoesjacabaramdepassarmasosinaldecaminholivreparaoscarrosvaitardarealgunssegundoshaquemsustentequeestademora
aparentementetaoinsignificantiseamultiplicarmospelosmilharesdesemaforosexistentesnacidadeepelasmudancassucessivasdastrescoresdecadaeumeaum
adascausasmaisconsideraveisdosengorgitamentosdacirculacaoautomoveluoengarrafamentossenquisermosusarotermocorrenteosinalverdeacendouseenfi
mbruscamenteoscarrosarrancarammaslogosenotouquenaotinhamarrancadotodosporigualoprimeirodafiladomeioestaparadodevehaveraliumproblemame
canicoqualqueroaceleradorsoltoaalavancadacaixadevelocidadesquesencravououumaavariadosistemahidraulicoblocagemdostravoesfalhadocircuitoelectri
cosequeanaosleacabousimplesmenteagasolinanaoseriaaprimeiravezquesedavaocaseonovoajuntamentodepeoesqueestaaformarsenospasseiosveocondu
tordoautomovelimobilizadoaesbracejarportrasdoparabrisasenquantooscarrosatrasdelebuzinamfreneticosalgunscondutoresjasaltaramparaaruadispostosa
empurraroautomovelempanadoparaondenaofiqueaestorvarotransitobatemfuriosamentenosvidrosfechadosohomemqueestaladentroviraacabecaparaeles
aumaladoaoutrovesquegritaqualquercoisapelosmovimentosdabocapercebesequerepeteumapalavraumanaoduasassimehrealmenteconsoantevaificarasa
berquandoalguemenfimconseguirabrirumaportaestoucego

Question 2

Para decifrar a cifra de vigenère utilizei um brute force até alguma palavra presente nos argumentos da chamada do script python estar na string final.
O script tem o seguinte flow:

1 - Faço o parsing dos argumentos do script vigenereattack.py:

```python
def parse_args():
    if len(sys.argv) < 4:
        print("Usage: python vigenereAttack.py <key_size> <ciphertext_file> <word1> [<word2> ...]
        ")
        sys.exit(1)

    try:
        key_size = int(sys.argv[1])
        if key_size <= 0:
            raise ValueError
    except ValueError:
        print("Error: key_size must be a positive integer.")
        sys.exit(1)

    ciphertext_file = sys.argv[2]
    try:
        with open(ciphertext_file, 'r') as f:
            ciphertext = f.read().strip().upper()
    except FileNotFoundError:
        print(f"Error: File '{ciphertext_file}' not found.")
        sys.exit(1)

    words = [word.upper() for word in sys.argv[3:] if word.strip()]
    if not words:
        print("Error: At least one non-empty word must be provided.")
        sys.exit(1)

    return key_size, ciphertext, words
```

Imagem 2.1: Função que faz parse dos argumentos do script

Assim teremos acesso dentro do código dos argumentos na chamada do script.

```
python3 vigenereattack.py 4 ciphertext.txt DIREITO FANTASIA LIBERDADE INTERVENCAO
```

Imagem 2.2: Chamada do script com argumentos

2 - A função de brute force é chamada. Dentro dela existem 2 loops principais e um auxiliar:

- Loop para rotação da chave: A chave poderá começar com k != 0, então é importante termos em conta esse corner case.
- Loop para iterar sobre todas as permutações possíveis de chaves (A-Z) com um tamanho especificado no argumento da chamada ao script.
- Loop para checar se uma palavra está presente na string final, após ser decifrada com uma iteração da chave.

```python
def brute_force_vigenere(key_size):
    for rotation in range(key_size):

        for key in itertools.product(range(len(chars)), repeat=key_size):

            # Rotate the key depending on the rotation value
            # [-rotation:] -> Get the last 'rotation' elements
            # + key[:-rotation] -> Add the rest of the elements from the start
            rotated_key = key[-rotation:] + key[:-rotation] if rotation else key
            decrypted = use_key(rotated_key)

            for ans in res:
                if ans in decrypted:
                    print(f"Key: {rotated_key} (rotation {rotation})")
                    print(decrypted)
                    return

    print("No valid key found.")
    return
```

Imagem 2.3: Função de Brute Force

3 - Em cada iteração do brute_force_vigenere a chave é utilizada em toda a mensagem, de forma a produzir uma string final, que será testada com uma palavra dada pelo atacante. Para utilizarmos a chave, é feito um loop na mensagem com a função decrypt_letter(letter, k), que decifra uma letra utilizando o método de vigenère para trás.

```python
def decrypt_letter(letter, k):
    return chars[(chars.index(letter) - k) % len(chars)]

def use_key(key):
    decrypted = ""

    for i in range(len(cyphertext)):
        decrypted += decrypt_letter(cyphertext[i], key[i % len(key)])
    return decrypted
```

Imagem 2.4: Função que decifra a mensagem

4 - É feito o brute force com todas as permutações possíveis e no final a mensagem decifrada é imprimida, junto com sua chave.

```
mutante  main -  python3 vigenereattack.py 4 ciphertext.txt DIREITO FANTASIA LIBERDADE INTERVENCAO
Key: (0, 6, 8, 17) (rotation 0)
CONSIDERANDOQUEORECONHECIMENTODADIGNIDADEINERENTEAMEMBROSDAFAMILIAHUMANAEDEDIREITOSIGUAISEINALIENAVESEOFUNDAMENTODAILIBERDADEDAJUSTICAEDAPAZNOMNDOMUNDOCONSIDERANDOQUEODESPRE
ZOEODESPEITOPELODIREITOSHUMANOSRESULTARAMEMATOSBARBAROSQUEULTRAJAREACONSCIENCIADAHUMANIDADEEQUEOADVENTOIDEUMMUNDOEMQUEMULHERESEHONENSGOZEMIBELIBERDADEDEPALAVRADECENCAEDAPILI
BERDADEVIVEMASALVODOTEMORDANECESSIDADEREMOSTROUPROPATOTAPAROECLAMADOCOMOAMAISALTAASPIRANAOEOSEROMANHOMUMNGUNSMUMSIDERNEDOSERESSENCHIALQUEADIREITOSHUMANOESSEJAMPROTEGIDOSHPEL
OIMPERIODALEIPARAGUEOSERUMANNAOSEJACOMPELIDOCCOMOULTIMOSECURSOAREBELIHAOECONTRAAOPRESSAOCONSIDERANDODISEREVENCIAPROMOVEDOSESTADUALIDEADECLARERELACOESAMISTOSASCENTREASCONSIDE
RNAANDOQUEOSPOVOSADANNASUNIDASRADAFIRMARAMNACARTASUAPEINNOSDIREADIREITOSMESETUNDAMENTAISGASERUMANONDIGNIDADENONOVALDAPESSOAHUMANAEANUALDADEDEMISDESUELDADEDOMEMDAMOMULEERDECI
DIRAPROMOVERPROGRASSESOCIALELCONDICOESVIDAEMUMALIBIRDEAMASAMPLACONSIDERANDQUEOSPAISESMEMBROSSECONTPROMETERAMAPROMOVEREMCONOPERACAOCOMASNACOESUODASORESEITEOUNIVERSALAOSDIREIT
OSDIBIRDASDEFUNDAMENTAISSERSHUMOAELSOBSERVANCIADESSESDIREITOSIBERDADESCONSIDERANDOQUEUMANCPRECENSAOCOMUNMDESSDIREITOSMOREMLIERDADEUDAMAISALTAIMPORTANCIAARAPLENOCKPMRIUMENOID
ESSECOMPROMISSAGAOSMPOTATOANASSEIDEIAERGALPROCLAMASPRESENTEDECLARANUNIVERSADUSHDIREITOSHUMANCONOIDEALCOMUMARESEATINGIDOPORITODOSPOVOSTODASNMCOESCONBJETIVDEQUECADAINDIVIDUECA
LGAORGSNSOCIDADETENDOSEMPREMMENTESTADECLARAESFORCERSEPORMEIRRDOENSINODAEDUCACAEPORPROMOVEORRESPEITAESSESDIREITOSELIAPERDADESEPELALADOCADDEMDIAPROGRESSIVASDECARATERNACIONALEI
NTERNACIOLIALPGRASSEGURAOSERECNHECIMENTOESUAOBSERVNCAUNVERSAISEETETTIVOSTANETRJOSPOVOSODPROPRSOPASSESMBRSGRANTOENTROSPOVOSODOSTERRRIOSIASPJURISDICAO
```

Imagem 2.5: Resultado final