Ex2: Quebrando Cifra por Transposição

O Que é a Cifra por transposição?

A cifra por transposição é um método de criptografia que reorganiza os caracteres do texto claro seguindo um padrão específico para criar o texto cifrado. Ao contrário das cifras de substituição, que alteram os caracteres individuais por outros, a cifra de transposição mantém os caracteres originais, porém os embaralha conforme uma chave ou esquema predefinido.

Funcionamento da Cifra de César

1. **Escolha de uma Chave**:
    - A cifra de transposição requer uma chave ou uma regra que determine como os caracteres serão reordenados. Para a cifra implementada, não é possível repetir caracteres e nem ter uma chave muito grande.

1. **Processo de Codificação**:
    - O texto a ser criptografado é disposto em colunas de acordo com o tamanho da chave, e o texto criptografado deve ser a ordem de colunas de acordo com a ordem alfabética da chave.

1. **Exemplo Simples**:
    - Texto simples: "odiaestamaravilhosohoje"
    - Chave: lindo
    - Texto cifrado: "amioadtaojosrhociavseealhb"

Vantagens e Limitações

- **Vantagens**:
    - Fácil de implementar e entender.
    - Quase impossível de se quebrar na força bruta para chaves grandes.

- **Limitações**:
    - Com textos muito longos, a complexidade computacional para realizar a transposição pode aumentar significativamente.
    - A eficácia da segurança depende fortemente da complexidade da chave utilizada para transposição. Chaves fracas podem ser facilmente quebradas.

Implementação

Criptografando a mensagem(cifraTranspo.py)

Para a criptografar a mensagem primeiramente o é perguntado ao usuário qual a chave desejada, essa chave deve ter apenas caracteres do alfabeto sem repetição e deve ter até no máximo 8 caracteres. Que então pega a mensagem escrita no arquivo mensagem.txt e para cada caracter realiza um mapeamento para cara coluna dado o tamanho de uma chave, completando a matriz com letras do alfabeto. Então o programa pega esse mapeamento e cria uma mensagem criptografada pela ordem alfabética da chave. Posteriormente ele salva essa mensagem criptografada em um arquivo chamado cypher.txt.

Decriptografando

- **Força Bruta (forceDescripTranspo.py)**:
  Mesmo com um alto número de combinação para uma grande chave, para chaves menores, a força bruta ainda é uma alternativa viável. Com isso primeiramente para descriptografar o código, podemos separar com base no tamanho da cifra os possíveis tamanhos chave, visto que o tamanho da chave deve ser divisor do tamanho da cifra. Com o tamanho da chave em mãos iremos separar a cifra em colunas e para cada possível tamanho, iremos verificar a permutação das colunas. A permutação que gerar a frase mais provável de ser uma frase da língua portuguêsa é provavelmente a mensagem original. A verificação é feita igual ao da cifra de deslocamento, através da probabilidade acumulada dos dígrafos.
- **Análise de frequência (descripTranspo.py)**:
  Para esse método, diferente da força bruta, utilizaremos uma análise com base na frequência de dígrafos. Inicialmente é idêntico ao de força bruta, separando em colunas e analisando os possíveis tamanhos de chaves, mas ao invés de simular todas as possibilidades, o código escolhe uma coluna e então calcula a probabilidade de cada uma das outras colunas serem vizinhas dessa escolhida inicialmente. Então o código escolhe a coluna mais provável, e agora tendo duas colunas como ponto referêncial à esquera ou a direita, o processo se repete até que todas as colunas tenham sido selecionadas assim formando a combinação mais provável de colunas para um determinado tamanho de chave. Por fim o código pega essas mensagens para cada tamanho possível de chave e escolhe o mais provável entre eles como resposta, utilizando a mesma verificação final da força bruta. Esse método apesar de não garantir que a mensagem original seja encontrada é extremamente mais eficiente que a força bruta, principalmente em chaves maiores visto que existem N! possíveis soluções (N sendo o tamanho da chave).