Implementação e Ataque da Cifra de Vigenère

1st Leonardo Alves Riether
Dep. Ciência da Computação
Universidade de Brasilia
Brasília, Brasil
190032413@aluno.unb.br

***Abstract*— [1] This document is a model and instructions for LaTeX. This and the IEEEtran.cls file define the components of your paper [title, text, heads, etc.]. *CRITICAL: Do Not Use Symbols, Special Characters, Footnotes, or Math in Paper Title or Abstract.**

***Index Terms*—component, formatting, style, styling, insert**

I. Introdução

II. Estruturação do Projeto

Neste trabalho, foi implementado um codificador e decodificador da cifra de Vigenère em C++. O projeto foi compilado com GCC 12.1.0 e testado em Linux , mas em princípio pode ser compilado em qualquer sistema com GCC que suporte C++17 e executado tanto em sistemas baseados em Unix quanto no Windows.

O projeto foi dividido em três partes:

1) **main:** onde estão implementadas as funcionalidades de cifração e decifração da cifra de Vigenère, dada uma chave (ou arquivo de chave). Esse módulo é explicado na seção III e pode ser compilado com o comando `make main`.
2) **findkey:** onde está implementada a função de ataque da cifra de Vigenère, que encontra uma chave automaticamente, dada uma mensagem cifrada. Esse módulo é explicado na seção IV e pode ser compilado com o comando `make findkey`.
3) **test:** possui alguns, poucos, testes, que podem ser executados com o comando `make test`.

III. Implementação

A. Operação de Mescla da Mensagem com a Chave

Tradicionalmente, na cifra de Vigenère cada letra da mensagem é ”mesclada” com uma da chave, por meio da soma módulo 26 dos valores das letras (geralmente, o valor de A é 0, o de B é 1, e assim por diante). Essa abordagem é interessante quando se deseja cifrar texto, por sua simplicidade de implementação. No entanto, existem algumas desvantagems:

- Espaços e pontuação não são cifrados, portanto é fácil descobrir o tamanho de cada palavra.
- É mais difícil cifrar imagens, vídeos e arquivos binários que existem hoje em dia.
- Existem apenas 26 possibilidades para cada letra da mensagem

Dado isso, neste trabalho foi implementada uma variação da cifra de Vigenère tradicional, que utiliza a operação de **xor bit-a-bit** entre cada byte da mensagem e da chave. Com isso, é mais difícil descobrir o tamanho das palavras de um texto cifrado, é possível cifrar qualquer tipo de arquivo, e existem 256 possibilidades para cada byte da chave, tornando essa cifra um pouco mais difícil de quebrar.

B. Codificação em Base 64

Uma desvantagem de utilizar o xor é que o arquivo cifrado pode conter caracteres ilegíveis. Para manter uma certa compatibilidade com a cifra de Vigenère implementada com soma módulo 26, foi implementado um codificador e decodificador de base 64. Os arquivos cifrados e codificados em base 64 utilizam apenas caracteres legíveis, em troca de um aumento de aproximadamente 33% do tamanho do arquivo.

C. Argumentos de Linha de Comando

Para a execução do programa, após compilar com `make main`, [TODO].

IV. Ataque

V. Conclusão

References

[1] Leonardo Alves Riether.