Projeto 1 - Criptografia

Gabriel V. C. Candido, Stephanie B. Americo, Talita H. C. Fernandes

11 de Junho de 2018

1 PLAFEVIW

A solução adotada para o Trabalho 1 do curso de Criptografia consiste em uma combinação de três métodos de cifras simétricas e a conversão para um arquivo de áudio no formato WAV. Desta forma, o método - denominado *Play Fence Vigekey WAV* (PLAFEVIW [pla.fê.ví]) - recebe como entrada três chaves e um texto claro, e devolve um arquivo em WAV que contém o texto cifrado.

O dicionário utilizado será os caracteres do intervalo $[32, 256)$ na codificação UTF-8, além de dois *caracteres lixo* fora desse intervalo.

1.1 Cifra

O PLAFEVIW aplica as cifras simétricas *Playfair*, *Rail Fence* e *Autokey Vigenère*, nesta mesma ordem. Cada cifra utiliza uma das três chaves de entrada, não havendo qualquer dependência entre os passos. O texto cifrado final é utilizado para compor um arquivo de áudio no formato WAV. Desta forma, o texto cifrado pode ser transmitido na forma de áudio ou ainda escondido em outras mídias (vídeos, músicas, etc).

O *caractere lixo* que cada método insere a cada passo pode comprometer a consistência do texto original, pois durante a decifra os *caracteres lixo* são interpretados de forma diferente por cada método e acabam alterando o texto de forma a torná-lo ininteligível. Para evitar esse resultado, o *caractere lixo* é diferente para cada um dos métodos. Desta forma, a cada passo da decifra o *caractere lixo* correspondente pode ser removido e apenas o texto limpo é entregue para o passo seguinte.

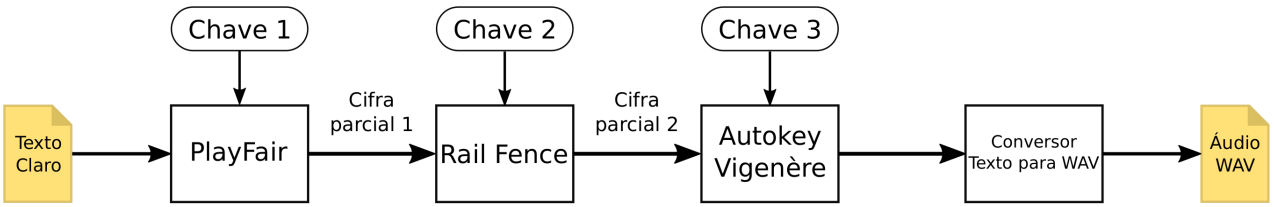

Figura 1: Esquema de cifra do texto claro.

1.2 Decifra

A decifra do PLAFEVIW segue exatamente o caminho oposto, visto que se trata de uma cifra simétrica. O única cuidado adicional é que cada método de decifra deve remover os *caracteres lixo* correspondentes ao seu passo, e passar o texto limpo para o seguinte.

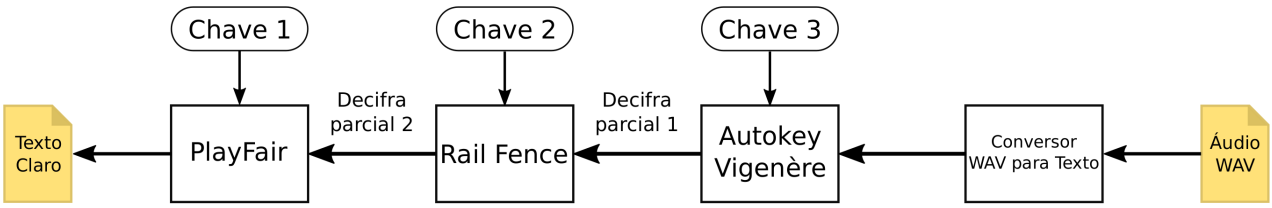

Figura 2: Esquema de decifra do texto cifrado.

2 Cifra Playfair

A cifra Playfair é uma cifra simétrica de substituição em blocos digrâmicos. As letras do texto claro são tomadas duas a duas, ao invés de uma a uma, como em outras cifras mais simples. Para a cifragem, originalmente, é utilizada uma matriz 5x5, que é criada tendo na primeira linha uma chave, e nas linhas subsequentes, as letras restantes do alfabeto em ordem, retirando-se as letras da chave (e I e J ocupam a mesma posição). Cada par de letras é codificado da seguinte forma:

- Se ambos estão na mesma linha, eles são trocados pela letra à sua direita.
- Se estão na mesma coluna, são trocados pela letra acima
- Se estão em linhas e colunas diferentes, cada letra é substituída pela letra na mesma linha e na coluna em que está a outra letra.

A vantagem da Playfair é evitar a análise por frequência de letras, pois o mapeamento entre as letras não é sempre o mesmo. Na cifra PLAFEVIW, ao invés de 5x5, será usada uma matriz 15×15, para comportar todos os caracteres relevantes da língua portuguesa.

2.1 Complexidade da Cifra

2.1.1 Complexidade de Espaço

Na Playfair é preciso uma matriz para armazenar todos os caracteres do alfabeto, portanto a complexidade de espaço é $O(|\Sigma|)$, onde Σ é o alfabeto utilizado. Como o alfabeto geralmente é fixo nas implementações (incluindo no nosso trabalho), essa complexidade é fixa.

Em nosso trabalho, a matriz terá tamanho $15 \times 15 = 225$. O algoritmo consome o texto de entrada em blocos de duas letras que não precisam ocupar espaço permanente em memória, podem ser cifrados e impressos imediatamente após a cifra.

2.1.2 Complexidade de Tempo

Para cada digrama é necessário percorrer a matriz da cifra Playfair para encontrá-los e só depois fazer a substituição. Portanto, a complexidade de tempo é $O(n \times |\Sigma|)$, sendo n o tamanho do texto e $|\Sigma|$ o tamanho do alfabeto e da matriz. Vale notar que $|\Sigma|$ pode ser visto como um valor quadrático do tamanho da matriz e pode crescer rapidamente com a inserção de novos caracteres no alfabeto. Aqui temos também o custo de criar a matriz, $O(|\Sigma|)$.

Pretendemos utilizar estruturas de dados que acessem elementos através de tabelas *hash*, o que possibilitaria um acesso rápido à matriz e reduziria a complexidade para $O(n)$. Ainda precisamos do custo para preencher a estrutura de dados, $O(|\Sigma|)$[1], mas para textos longos o suficiente, prevalece $O(n)$.

[1] Apesar de ter características aparentemente quadráticas para ordenar e verificar qual elemento já foi incluso, essas estruturas de dados permitem uma inserção rápida também.

Essa cifra é a única das três utilizadas (nesse modelo) que pode ser paralelizada para diminuir o tempo de execução da cifra/decifra.

3 Cifra Rail Fence/Transposição Colunar

A Rail Fence é uma cifra de transposição. Escrevemos o texto em uma matriz com um número de colunas determinado por uma palavra-chave. Cada letra da chave é substituída por um número segundo as seguintes regras: a letra mais próxima do começo do texto recebe o número 1, a segunda mais próxima o número 2 e assim por diante.

Caso haja duas letras repetidas, a mais próxima do começo do texto recebe o número menor, que vai incrementando para cada repetição. As colunas da matriz são numeradas conforme estes números. O texto cifrado é criado lendo-se a matriz por colunas, em ordem crescente a partir da numeração das colunas.

A vantagem da Rail Fence é evitar a análise por digramas, pois o texto cifrado é embaralhado e não mantém essa característica.

3.1 Complexidade da Cifra

3.1.1 Complexidade de Espaço

Como precisamos ler o texto todo para conhecer o seu tamanho e fazer a construção da matriz, temos complexidade $O(n + |k|)$, com n sendo o tamanho do texto e $|k|$ o tamanho da chave. Por mais que a matriz tenha um tamanho variado conforme a chave, ela irá se estender apenas o suficiente para comportar o texto da entrada.

3.1.2 Complexidade de Tempo

Precisamos percorrer o texto uma vez para descobrir o seu tamanho e organizá-lo na matriz, caso uma seja usada, e depois novamente para gerar o texto cifrado. Portanto, a complexidade é $O(2n) = O(n)$, onde n é o tamanho do texto.

4 Cifra de Vigenère

A cifra de Vigenère é uma cifra de substituição polialfabética que consiste da escolha de uma cifra de César para ser usada na letra corrente a partir de uma chave. Essa escolha se dá pela disposição das cifras de César com diferentes deslocamentos em uma matriz. Conforme o caractere da entrada e o caractere corrente da chave, um caractere é escolhido para compor o texto cifrado. Desta forma, a análise de frequência de letras não é efetiva para esse método.

O método utilizado é conhecido como *Autokey Vigenère*, onde a chave é utilizada nos primeiros caracteres a serem cifrados e posteriormente os próprios caracteres criptografados são usados como chave para caracteres subsequentes. Esse método auxilia a evitar que digramas sejam identificados no texto cifrado, uma vez que a chave seria repetida ao longo do texto.

4.1 Complexidade da Cifra

4.1.1 Complexidade de Espaço

Como o texto é cifrado caractere a caractere, não precisamos de espaço para guardar a entrada em memória (apenas o caractere a ser lido). Por outro lado, precisamos guardar a tabela de Vigenère, com todas os deslocamentos possíveis da cifra de César. Nesse caso, como temos

um alfabeto de 225 caracteres (intervalo de $[32, 256) + 2$), precisaríamos de uma tabela com $225^2 = 50625$ caracteres.

Contudo, a cifra de Vigenère pode ter seu deslocamento calculado algebricamente. Visto que a entrada é mapeada para os inteiros contíguos de 32 a 255 e que para o caractere k_i na i-ésima posição da chave k o primeiro caractere do dicionário é mapeado para o próprio k_i, temos que:

$$C_j = (k_i - \Sigma_0) + P_j$$

onde:

- P_j é o j-ésimo caractere do texto claro;
- k_i é o inteiro associado ao i-ésimo caractere da chave k;
- Σ_0 é o inteiro associado ao primeiro caractere do nosso dicionário; e
- C_j é o caractere cifrado na posição j

De forma análoga pode ser feita a decifra, dispensando assim a necessidade de manter a tabela em memória. Precisamos apenas guardar a chave, que pode ser sobrescrita gradualmente conforme os novos trechos da chave são gerados pelo algoritmo. Temos complexidade $O(|k|)$, o tamanho da chave.

4.1.2 Complexidade de Tempo

A cifra de Vigenère percorre todo o texto uma vez aplicando os devidos deslocamentos, portanto a complexidade é $O(n)$, onde n é o tamanho do texto.

5 Complexidade Total da Cifra

Agregando a complexidade de cada cifra, temos que a complexidade de espaço é da ordem $O((|\Sigma|) + (n + |k_{playfair}|) + (|k_{vigenere}|))$. Esse valor, para textos grandes, deve convergir para $O(n)$.

A complexidade de tempo é da ordem de $O(n + 2n + n) = O(4n) = O(n)$.

6 Complexidade da Quebra

Partindo do princípio que, em conjunto, as cifras utilizadas removem todas as frequências que podem ser analisadas no texto, as possibilidades de quebra do algoritmo seriam parciais em cada uma das cifras.

Se houver posse dos textos intermediários da Rail Fence, as posições das letras podem ser comparadas para encontrar lixos e descobrir a chave da cifra, mas como usamos o método com colunas numeradas, essa análise se torna mais complicada.

Acreditamos que dessa forma é possível quebrar a Playfair com o texto intermediário, analisando digramas do alfabeto. Já Vigenère, como complementa a chave conforme ocorre a cifra, pode ser mais difícil de quebrar através de análise.

Com a combinação dos métodos e sem a posse dos textos intermediários, acreditamos que as análises estatísticas sobre os textos são impossibilitadas.

A quebra, por força bruta, envolve:

- $|\Sigma|!$ matrizes diferentes para a Playfair;
- n larguras de linha para a RailFence; e
- $|\Sigma|^n$ testes para Vigenère: em cada caractere todos os símbolos do dicionário são testados.

Isso totaliza $|\Sigma|! \times |\Sigma|^n \times n$ tentativas para quebrar a cifra PLAFEVIW por força bruta. Em nossa implementação, temos $255! \times 225^n \times n =$ possibilidades.

Referências

[1] Notas de Aula de Criptografia. Luiz C. P. Albini, Ivan L. P. Pires.
`http://www.inf.ufpr.br/albini/tutorial_cripto/`

[2] Departamento de Ciência de Computadores da F.C.U.P.
`https://www.dcc.fc.up.pt/ nam/aulas/9900/pi/trabalho4/cripto.html`

[3] Transposition cipher. Wikipedia.
`https://en.wikipedia.org/wiki/Transposition_cipher`

[4] Cifra de Vigenère. Wikipedia.
`https://pt.wikipedia.org/wiki/Cifra_de_Vigenère`