Trabalho 1 Segurança de Sistemas

Victor B. Diehl

Curso de Bacharelado em Engenharia de Software – Pontifícia Universidade Católica do Rio Grande do Sul (PUCRS)
Porto Alegre – RS– Brasil

victor.diehl@acad.pucrs.br

Resumo. *Este meta-artigo descreve o primeiro trabalho para a disciplina de segurança de sistemas do curso bacharelado em Engenharia de Software da Escola Politécnica da PUCRS. Nele iremos desenvolver e apresentar um programa capaz de decifrar um texto cifrado utilizando o método de* índice de coincidência para descobrirmos a chave e, posteriormente, adquirir o texto claro.

Abstract. *This article describes the first work for systems security discipline of baccalaureate courses in Software Engineering of the Polytechnic School of PUCRS. We will develop and present a program able to decipher a ciphertext using the matching index method to discover a key and then get the real text.*

1. Introdução

Existem várias maneiras de decriptar textos cifrados. A raiz de problemas deste gênero é a chave utilizada para criptografar determinado texto ou código. Um exemplo para descobri-la é o teste de Friedrich Kasiski na qual procuramos segmentos idênticos e encriptados pela mesma substring, e a partir disso sabemos que suas distâncias são múltiplos do tamanho da chave, na qual pode ser descoberta pelo maior divisor comum. Em compensação neste artigo iremos apresentar outro método chamado de índice de coincidência.

Essa lógica baseia-se em utilizar a frequência dos caracteres de um pré-determinado alfabeto (português, inglês, espanhol, etc...) e realizar uma comparação com o índice de coincidência dos caracteres do texto encriptado. Na próxima seção vamos explicar, em trechos, detalhadamente como esse processo é realizado utilizando a linguagem de programação Swift para contextualizar sua execução em passos.

2. Desenvolvimento

Existem quatro principais etapas que devem ser executadas na ordem para atingirmos a solução do problema:

1. Definir a linguagem usada para verificar a coincidência e ordená-la de acordo com os caracteres mais frequentes (normalmente são realizadas tentativas com diferentes linguagens mas neste caso sabemos que o texto encriptado foi escrito em português).
2. Descobrir o tamanho da chave de segurança.
3. Descobrir os caracteres da chave de segurança.

4. Traduzir o texto com a chave descoberta.

Cada subseção abaixo irá desenvolver os passos para chegarmos a solução. Optamos por não exibir o primeiro passo já que ele simplesmente contém o inserção e a ordenação crescente de frequência dos caracteres da língua portuguesa (["a": 14.634%, "e": 12.510%, "o": 9.735%, etc...]).

2.1 Descobrir Tamanho da Chave

Para descobrirmos o tamanho da chave utilizamos a técnica de divisões em cosets referentes a junção de caracteres de mesma distância. Por exemplo se possuímos o texto **PVCSSA** e utilizarmos 2 como o valor de distância teremos dois cosets, **PCS** e **VSA**, enquanto com 3 de valor teremos **PS**, **VS** e **CA**. O texto no Apêndice é claramente maior que esse exemplo portanto cada coset é maior também.

Com eles definidos para um valor, podemos aplicar a função que verifica o índice de coincidência dos caracteres com o a língua portuguesa. Essa operação é feita através da contagem das ocorrências de cada caracter do coset e comparadas com o texto encriptado original. Ao final podemos fazer a média da soma dos resultados de coset e teremos nosso índice de coincidência para a distância verificada. O ideal é que este procedimento seja feito com diferentes distâncias pois sempre estaremos em busca de valores próximos a 100% equivalente ao tamanho da chave.

2.2 Descobrir Caracteres da Chave

Após o passo anterior e agora com o tamanho da chave, podemos descobrir qual caracter equivale a cada posição. Em um primeiro momento aplicamos o conjunto coset, da distância vitoriosa, sobre a técnica chamada shiftLeft para reposicinoar os caracteres em suas respectivas posições de acordo com o texto encriptado para reverter o passo realizada criação do texto.

A operação subsequente a esta, compara a frequência de cada caracter com todos da lingua portuguesa. Esta operação também conhecida como X^2, aplica a fórmula $X^2 = \sum_{i=1}^{n} (f_i - F_i)^2 / F_i$ tal que f_i é o valor frequência do caracter observado enquanto F_i é a respectiva frequência do valor na linguagem. Essa é uma função que tende a 0, quando menor o valor do resultado do somatório, maior a chance de ele estar correto. Esta etapa exige certo processamento diante da comparação de todos para todos mas com ela obtemos o valor correspondente da chave.

2.3 Traduzir Texto com Chave

Por fim, para deciframos o texto, é realizado o processo inverso, os caracteres são deslocados para a esquerda no alfabeto de acordo com a chave.

3. Resultado

Baseando-se no exemplo do apêndice, em um primeiro momento utilizamos uma iteração de 2 a 20 de distância na fase de busca pelo tamanho da chave. Grande parte do cosets atingiram 40% de precisão, mas utilizando cosets de 7 posições de distância, atingimos 76,39%. Dessa forma descobrimos que este era o tamanho da chave.

Já na segunda etapa utilizamos este coset de 7 posições de distância aplicando o shiftLeft sobre o texto encriptado e a função de comparação das frequências, conseguindo o valor da chave = "avelino".

No final utilizamos o quanto a posição de cada caracter da chave foi deslocada para coloca-lo no local correto o deslocando para esquerda. Na seção de anexo existe um link do github de acesso ao código contendo o texto encriptado e decifrado.

4. Conclusão

Este trabalho permitiu adquirir um enorme conhecimento, não só em relação a métodos de decriptografia como o índice de coincidência que agiram como foco deste desenvolvimento, como também, sobre a teoria por trás das criptografias textuais e seu nível de complexidade. Conseguimos decifrar o texto desta maneira descrita no artigo mas reconhecemos que melhorias podem ser feitas para que o tempo de execução seja reduzido.

O artigo foi redigido buscando uma forma simples para transmitir este conhecimento para aqueles que trabalham no desenvolvimento de softwares no cotidiano, como também aos leitores que estão simplesmente interessados no conteúdo e possam interessar como um passo inicial ao ramo mesmo essa sendo uma criptografia básica.

5. Referências

1. https://pages.mtu.edu/~shene/NSF-4/Tutorial/VIG/Vig-Recover.html

6.Anexo

github: https://github.com/victordiehl/Seguranca-Sistemas-t1