

GBC083 – Segurança da Informação

Aula 1 - Introdução à Criptografia - Ataque à cifra de Vigenère

Prof. Marcelo Keese Albertini

Revisão

- ▶ Princípio de Kerckhoff
- ▶ Princípio do tamanho de espaço de chaves

Ataque à cifra de Vigenère

- ▶ Supondo que $|k| = 14$, de 14 em 14 caracteres usamos o mesmo deslocamento

veqjiredozxoeualpcmsdjquiqndnossoscdcusoakjgmxpqr

- ▶ Se pegarmos de 14 em 14 caracteres fica parecido com cifra de deslocamento, mas força-bruta não funciona

Usando frequências de letras em texto em claro

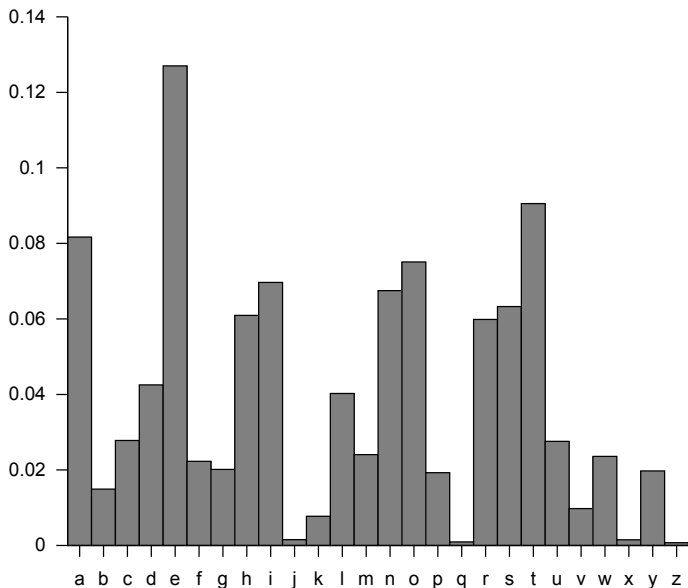


Figura: Fonte: [https://pt.wikipedia.org/wiki/Ficheiro:](https://pt.wikipedia.org/wiki/Ficheiro:English_letters_frequencies.png)

English_letters_frequencies.png

[CC BY-SA 4.0](#)

Atacando a cifra de Vigenère

- ▶ Olhar a cada 14 caracteres do texto cifrado iniciando na primeira posição da string
- ▶ Seja α o caractere mais comum nessa parte do texto cifrado
- ▶ Provavelmente α vai ser o equivalente à letra mais frequente do inglês 'e'
- ▶ Então a primeira letra da chave é $\alpha - 'e'$
- ▶ Repetir para as outras posições

Atacando a cifra de Vigenère

- ▶ A chave é uma string de letras
- ▶ Para encriptar, deslocar cada caractere no texto em claro pela quantidade definida pelo próximo caractere da chave
 - ▶ Dar a volta na chave quando necessário
- ▶ Decifração reverte o processo
- ▶ `tellhimaboutme` ⊗ `cafecafecafeca` = `veqojiredozxoe`

Variante da cifra de Vigenère

- ▶ Melhor trabalhar com texto em claro em ASCII e texto cifrado em hexa
 - ▶ Mais fácil para implementar
 - ▶ Mais fácil de usar – texto em claro não limitado a caracteres minúsculos
- ▶ Mais fácil trabalhar com XOR bit-a-bit que com adição modular

Variante da cifra de Vigenère

- ▶ A chave é uma string de bytes
- ▶ Texto em claro é uma string de caracteres ASCII
- ▶ Para encriptar, XOR cada caractere no texto em claro com o próximo caractere da chave
 - ▶ Dar a volta na chave quando necessário
- ▶ Decriptação só reverte o processo

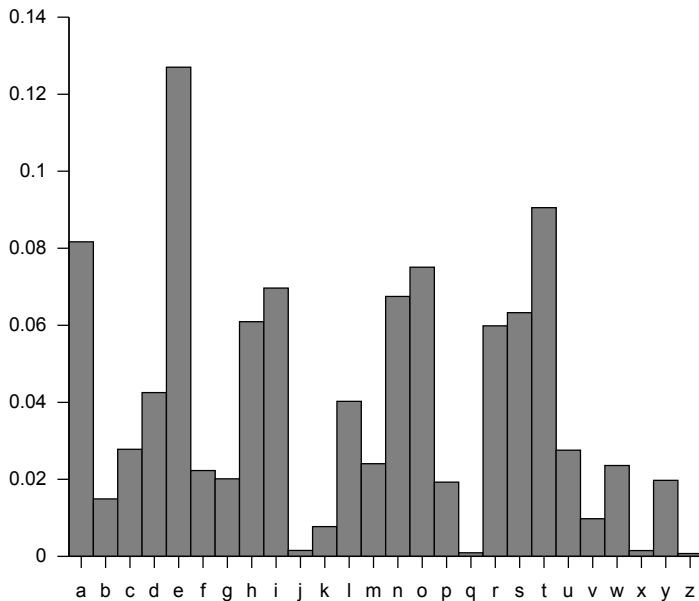
Exemplo

- ▶ Digamos que texto em claro é “Hello!” e chave é $0xA12F$
- ▶ “Hello!” = $0x48656C6C6F21$
- ▶ XOR com $0xA12FA12FA12F$
- ▶ $0x48 \oplus 0xA1$
 - ▶ $01001000 \oplus 10100001 = 11101001 = 0xE9$
- ▶ Texto cifrado: $0xE94ACD43CE0E$

Atacando a variante da cifra de Vigenère

- ▶ Dois passos
 - ▶ Encontrar o comprimento da chave
 - ▶ Determinar cada byte da chave

Frequências de letras de texto em claro



Encontrar o tamanho da chave

- ▶ Seja p_i , com $0 \leq i \leq 255$, a frequência do byte i em texto em claro em inglês
 - ▶ Assim, $p_i = 0$ se $i < 32$ ou $i > 127$
 - ▶ $p_{97} =$ frequência de 'a'
 - ▶ A distribuição não é uniforme
- ▶ Se tamanho da chave é N , então todo N -ésimo caractere em texto em claro é encriptado com o mesmo deslocamento
 - ▶ Extrair cada N -ésima letra e calcular frequência. Resulta em uma permutação de p_i
 - ▶ Se extrair cada M -ésima letra tal que M não é múltiplo de N e calcularmos frequências, o resultado é algo mais uniforme que p_i

Encontrar o tamanho da chave

- ▶ Como distinguir quando é permutação de p_i ou quando é uniforme?
- ▶ Para alguma distribuição candidata q_0, \dots, q_{255} computar $\sum q_i^2$
 - ▶ Se q for quase uniforme $\sum q_i^2 \approx 256 \cdot (1/256)^2 = 1/256$
 - ▶ Se q for uma permutação de p então $\sum q_i^2 \approx p_i^2$
 - ▶ Poderíamos computar $\sum p_i^2$, mas não é preciso
- ▶ Tentar todas as possibilidades de tamanho de chave e computar $\sum q_i^2$ e procurar pelo maior valor

Descobrir cada i -ésimo byte da chave

- ▶ Assumir que o tamanho da chave é N
- ▶ Extrair cada N -ésimo caractere do texto cifrado a partir do i -ésimo caractere
 - ▶ Esta é a i -ésima sequência do texto cifrado
 - ▶ Note que todos os bytes nessa sequência foram cifrados por XOR usando o mesmo byte da chave
- ▶ Tentar decifração com todo valor de byte B
 - ▶ Obter uma sequência de texto em claro candidata para cada valor B
 - ▶ Qual é o valor correto de B

Descobrir cada i -ésimo byte da chave

- ▶ Quando valor da chave B está correto:
 - ▶ Todos os bytes estarão entre 32 e 127
 - ▶ Frequências de letras minúsculas devem ser próximas às frequências conhecidas p
- ▶ Para achar B :
 - ▶ Contar q_a, q_b, \dots, q_z
 - ▶ Deve achar $\sum q_i p_i \approx \sum p_i^2 \approx 0.078$ ou (0.065 para o inglês)
- ▶ Na prática, obter B que maximiza $\sum q_i p_i$ e sujeito a condições de validade de texto (por exemplo, poucas vírgulas)

Tempo de ataque

- ▶ Seja N tamanho da chave entre 1 e L
- ▶ Sendo o conteúdo de tamanho constante k
- ▶ Para encontrar tamanho da chave:
 - ▶ Computar $\max \sum_{i=0}^{255} q_i^2$ variando $N \in [1, L]$
 - ▶ $\approx 256L \times k$
- ▶ Descobrir cada byte da chave:
 - ▶ Para cada valor em $[0, 255]$ obter novo q e fazer $\sum_{i=0}^{255} q_i p_i$
 - ▶ $\approx 256^2 N \times k$
- ▶ Busca pela chave por força bruta:
 - ▶ Se $N = 1$, número de chaves: 256
 - ▶ Se $N = 2$, número de chaves: 256×256
 - ▶ Se $N = 3$, número de chaves: 256^3
 - ▶ Se $N = L$
 - ▶ $\approx 256^L \times k$

O ataque na prática

- ▶ Ataque é mais confiável para sequências longas
- ▶ Ataque funciona para textos cifrados mais curtos mas necessita de ajustes manuais

Exercícios

- ▶ Implementar ataque à cifra de Vigenère
- ▶ Descobrir o conteúdo da mensagem em `http://www.facom.ufu.br/~albertini/infosec/texto-cifrado.txt`