Criptografia

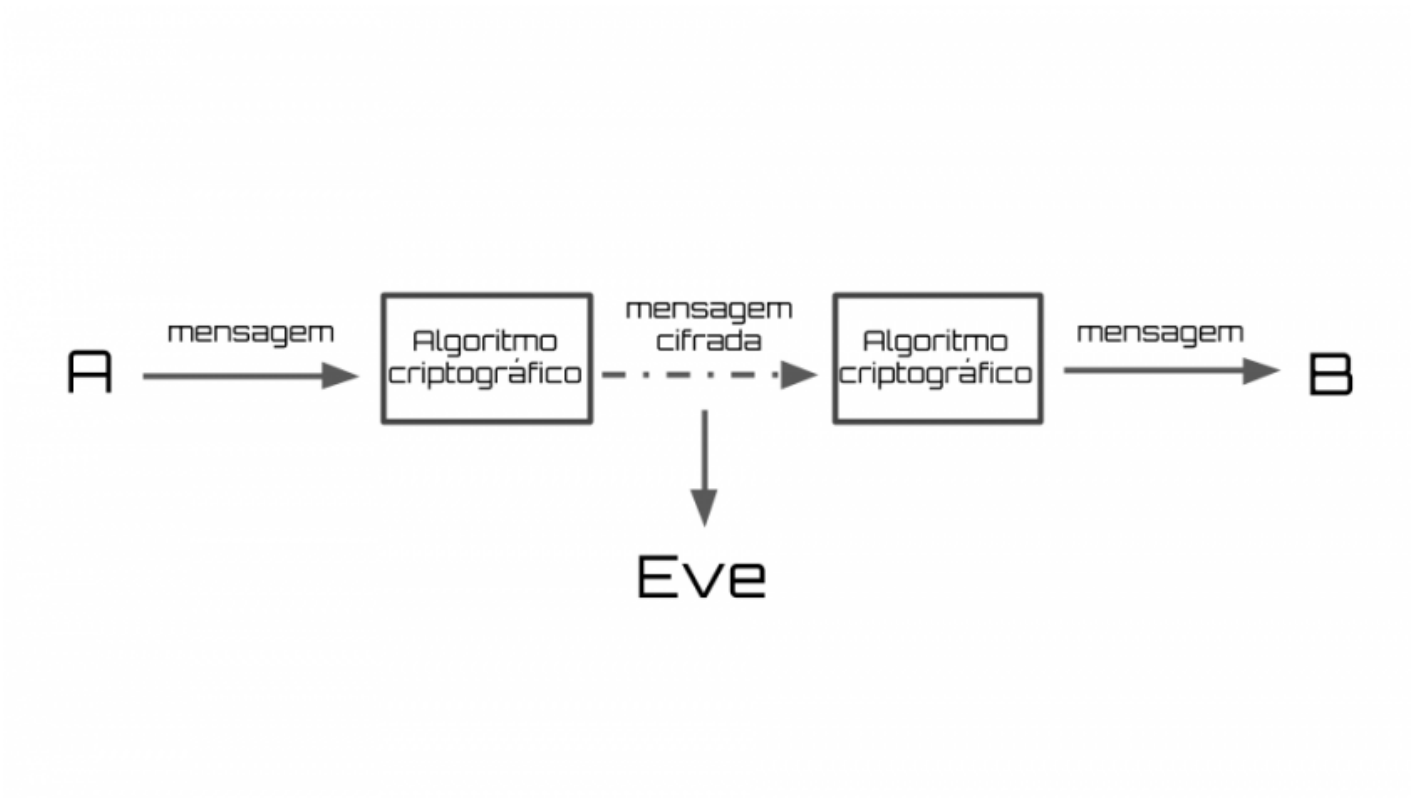
Guia sobre Criptografia

O que é criptografia?

Criptografia vem do grego *kryptós* e *graphein*, que significam "secreta" e "escrita", respectivamente. Até a era moderna, ela era sinônimo de *encriptação*, que é a conversão de uma mensagem legível para algo aparentemente sem sentido.

Para entender melhor essa ideia, digamos que **Alice** quer mandar uma mensagem para **Bob** sem que **Eve** descubra seu counteúdo.

Para isso, **Alice** usa um certo algoritmo para tornar a mensagem ilegível de forma que só **Bob** sabe como reverter a mensagem encriptada.



Assim, caso **Eve** consiga a mensagem ela, sem saber o algoritmo que **Alice** usou, não será capaz de entender a mensagem.

Ao longo da história, várias técnicas de ocultar mensagens foram desenvolvidas. Antes da criptografia pré-computacional, a **criptografia clássica** era formada por um conjunto de métodos de *substituição* e *transposição* de caracteres, porém com o advento da computação, a **criptografia moderna** se tornou amplamente embasada em teorias matemáticas e práticas de

ciência da computação.

Cifras antigas

Cifra de César

A Cifra de César é um dos métodos mais simples e comuns de encriptação. Mesmo não sendo muito comum em CTFs, ainda é um conhecimento básico de criptografia.

> “ Esse método tem esse nome pois era usado por Júlio César em suas correspondências

Nessa cifra, cada letra da mensagem é substituida por uma letra do alfabeto deslocado por um número fixo.

Por exemplo, se queremos encriptar a mensagem `hack the planet`, podemos deslocar cada letra do alfabeto **3 vezes para direita** (ou **right 3**). Assim, a substituição teria esse formato:


```
texto original: hack the planet


texto cifrado:  kdfn wkh sodqhw
```

Dessa forma, o texto gerado se torna incompreensível de forma que só quem sabe o algoritmo usado poderá recuperá-lo.

ROT13

Um dos tipos mais comuns de Cifra de César é o **ROT13**. Nele, o alfabeto é deslocado 13 vezes. Como o alfabeto tradicional possui 26 letras, o ROT13 possui a propriedade de que o mesmo algoritmo usado para encripitar a mensagem é usado para decriptar.

Detectando

Mensagens encriptadas pela cifra de césar normalmente produzirão um amontoado de caracteres sem significado, como `kdfn wkh sodqhw`, e suas letras terão uma distribuição de frequência similar à língua usada (provavelmente inglês), mas com as letras trocadas. Esse conceito será abordado com mais profundidade em Cifras de Substituição.

Devido a facilidade de quebrar essa cifra, pode ser conveniente tentar solucioná-la sem nem ao menos uma análise de frequência.

Solucionando

Como num alfabeto usual são usados apenas 26 caracteres, a Cifra de César possui apenas 25 tipos de rotações possíveis (pois a rotação 26 é a própria mensagem). Assim, um **testa tudo**, onde você faz todos os tipos de rotações possíveis, é a opção mais simples.

Existem ferramentas online muito eficientes para quebrar uma Cifra de César, como o site dcode, porém não é muito difícil codificar um *testa tudo* para isso.

Codificando um testa tudo

Primeiro, codificaremos uma função `rot()` que aplica a rotação em um caractere, de acordo com o deslocamento determinado (o `shift`):

```
```

```
def rot(char, shift):
  return chr((ord(char) - ord('A') + shift)%26 + ord('A'))
```

Assim, podemos usar essa função para criar um `caesar_brute_force()` que recebe um texto cifrado e imprime todas as rotações possíveis.

```
def caesar_brute_force(cipher_text):
    cipher_text = cipher_text.upper()
    for i in range(26):
        line = ''
        for c in cipher_text:
            line += rot(c, i) if c.isalpha() else c
        print(f'rot{i}:\t{line}')
```

Cifras de Substituição Simples

Em uma cifra de substituição simples, cada letra é substituida individualmente de acordo com um **alfabeto de substituição**. Esse alfabeto pode ser uma rotação fixa do alfabeto normal (como a cifra de César) ou algum embaralhamento mais complexo.

Alguns exemplos notáveis de cifra de substituição simples são:

Cifra de Atbash

> “ Seu nome tem origem da primeira, última, segunda e penúltima letra Hebraica (Aleph-Taw-Bet-Shin)

Nessa cifra, cada letra é mapeada para o alfabeto invertido, ou seja, a primeira vira a última, a segunda vira a penúltima e assim por diante.

```
original: A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
cifra:    Z Y X W V U T S R Q P O N M L K J I H G F E D C B A
```

Assim, se usarmos essa cifra em `may the force be with you`, obteremos:

```
original: M A Y  T H E  F O R C E  B E  W I T H  Y O U
cifrado:  N Z B  G S V  U L I X V  Y V  D R G S  B L F
```

A Cifra de Atbash pode ser interpretada como um caso particular da Cifra de Affine, uma cifra que usa aritimética modular para encriptar.

Cifra da Palavra-Chave

A Cifra da Palavra-Chave ou *keyword cipher* consiste em escolher uma **chave** e usá-la para decidir como as letras serão susbtituidas.

As palavras repetidas dessa chave serão removidas e a própria chave será o começo do alfabeto a ser mapeado. O resto das letras continuarão em ordem alfabética, tirando as letras já usadas.

Por exemplo, escolhendo a chave `Marvin`, o novo alfabeto terá esse formato

```
original: A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
cifra:    M A R V I N B C D E F G H J K L O P Q S T U W X Y Z
```

Assim, ao encriptar a mensagem `Arthur Dent`, obteremos:

```
original: A R T H U R  D E N T
cifrado:  M P S C T P  V I J S
```

Detectando

Como mencionado na seção de Cifra de César, uma mensagem encriptada por uma cifra de substituição simples terá uma distribuição de frequência das letras semelhante ao da língua usada, mas com as letras trocadas.

Essa distribuição de frequência de um texto pode ser identificada através de uma **análise de frequência**.

Nas línguas naturais, algumas letras aparecem mais frequentemente que outras, como uma espécie de digital do idioma. Por exemplo, a letra mais comum na lígua inglesa é o **"e"**, em português é o **"a"**.

Essa análise de frequência pode ser feita simplesmente contando as letras do texto. Existem ferramentas online para isso como o site dcode ou pode ser feito rapidamente com um biblioteca em Python, onde `text` é o texto a ser analisado:

```
from collections import Counter
Counter(text.upper()).most_common()
```

Solucionando

O ponto fraco de cifras de substituição simples é que elas são muito suscetíveis à **análises de frequência**.

Assim, se você tiver um texto de tamanho razoável, por volta de 50 caracteres, é possível analisar a frequência com que as letras aparecem e deduzir qual foi o alfabeto de substituição usado.

O site guaballa é um excelente decodificador de cifras de substituição simples.

Referências

Cifra de Atbash: Jeremiah's Game

Cifra da Palavra-Chave: GeeksforGeeks

Learn Cryptography

Cifras antigas

Cifra de Vigenère

Devido à vulnerabilidade das cifras de substituição simples, foi necessário a criação de uma cifra que conseguisse se proteger disso. A **Cifra de Vigenère** veio com esse propóstio e é basicamente uma extensão da fórmula da Cifra de César. Ela gera uma distribuição praticamente uniforme em uma análise de frequência e foi considerada **inquebrável** por 3 séculos.

> “ Ela tem esse nome em homenagem a Blaise de Vigenère

Essa Cifra consiste basicamente em pegar uma **palavra-chave** e aplicar a cifra de César várias vezes, de acordo com os caracteres da palavra-chave.

Por exemplo, se nós queremos encriptar a mensagem `the cake is a lie` usando a palavra-chave `portal`, primeiro cada caractere da palavra-chave terá um número de rotações equivalente (de acordo com sua posição no alfabeto):

etr	**P**	**O**	**R**	**T**	**A**	**L**
rotação	16	15	18	20	1	12

Assim, para cada letra da mensagem será rotacionada de acordo com a sequência de rotações acima:

```
mensagem:          T H E  C A K E  I S  A  L I E
chave:             P O R  T A L P  O R  T  A L P
mensagem cifrada:  I V V  V A V T  W J  T  L T T
```

Essa cifra, diferentemente das cifras de substituição simples, é uma **Cifra de Substituição Polialfabética**.

Detectando

Um texto encriptado por essa cifra pode ser detectado através de uma **análise de frequência**.

A Cifra de Vigenère costuma gerar textos com uma distribuição de frequência das letras próximo ao uniforme. Se um texto cifrado que não é esperado esse tipo de distribuição obter esse resultado, provavelmente é Cifra de Vigenère, ou alguma outra Cifra Polialfabética.

Solucinando

Mesmo gerando uma distribuição uniforme em análises de frequência, essa cifra tem uma vulnerabilidade: a palavra-chave é usada várias vezes em um texto grande.

Dessa forma, se a chave tiver tamanho 5, por exemplo, e ajustarmos o texto em linhas de comprimento 5, cada coluna terá a mesma rotação. Assim, podemos chutar tamanhos da palavra-chave e usar a mesma análise de cifra de substituição simples para cada coluna.

Uma ferramenta online muito útill para quebrar a Cifra de Vigenère é o site dcode.

Referências

GeeksforGeeks

Cifras de Transposição

Na tentativa de encontrar um método alternativo às cifras de substituição simples, que estavam se tornando frágeis, foram criadas as **Cifras de Transposição**. Nessa cifra, o texto permanece o mesmo mas as ordem dos caracteres são alteradas, embaralhando a mensagem de acordo com um padrão.

Existem vários padrões diferentes para realizar a transposição, os dois mais famosos são a **transposição colunar simples** e a **rail fence**.

Transposição colunar

Com essa regra, a mensagem é escrita horizontalmente numa matriz de largura fixa e a saída é o texto lido verticalmente nessa matriz.

Numa **transposição colunar simples** essa leitura é feita por colunas da esquerda para direita. Por exemplo, com texto `a wizard is never late` a encriptação será da forma

```
Texto: A WIZARD IS NEVER LATE

Matriz:

A W I Z A R
D I S N E V
E R L A T E

Texto cifrado: ADEWIRISLZNAAETRVE
```

Para decriptar, escrevemos o texto cifrado verticalmente numa matriz de mesma largura e lemos o texto horizontalmente.

A ordem de leitura de colunas pode ser determinada também de acordo com uma **palvra-chave**.

O tamanho da palavra-chave definirá a largura da matriz usada e cada caractere determina a ordem que as colunas serão lidas. Por exemplo, usando a chave `FRODO`, numeramos as letras em ordem alfabética: `25314`. Assim, podemos usar essa ordem para ler as colunas e gerar o texto cifrado

```
Texto:  A WIZARD IS NEVER LATE

Ordem:   2 5 3 1 4

Matriz:  A W I Z A
         R D I S N
         E V E R L
         A T E

Texto cifrado:  ZSRAREAIIEEANLWDVT
```

Rail Fence

Nesse tipo de transposição, os caracteres são escritos numa matriz usando um padrão fixo em zigue-zague e a saída é o texto lido horizontalmente. o rail fence admite várias variações, como a linha que a primeira letra começa e o número de linhas usadas. Por exemplo, usando um rail fence com duas linhas

```
Texto:  A WIZARD IS NEVER LATE

Matriz:

A - I - A - D - S - E - E - L - T -
- W - Z - R - I - N - V - R - A - E

Texto ciffrado:  AIADSEELTWZRINVRAE
```

Outro exemplo mas com três linhas será da forma

```
Texto:  A WIZARD IS NEVER LATE
```

```
Matriz:

A - - - A - - - S - - - E - - - T -
- W - Z - R - I - N - V - R - A - E
- - I - - - D - - - E - - - L - - -

Texto ciffrado:  AASETWZRINVRAEIDEL
```

Para descriptografar, é necessário o conhecimento do padrão usado e preencher as lacunas com o texto cifrado horizontalmente e depois ler em zigue-zague.

Essa técnica foi usada na Guerra Civil norte-americana para cifrar as mensagens dos confederados e dos federalistas.

Identificando

Como apenas a ordem do texto é alterada, a distribuição de frequência das letras será muito parecida com a frequência da língua usada. Assim, uma análise de frequência do texto cifrado é um ótimo método para identificar o uso de uma cifra de transposição.

Além disso, podem ter sido usadas cifras de substituição em conjunto, dificultando a indentificação.

Solucionando

Um primeiro método que podemos pensar para quebrar cifras de transposição é testar todas as possíveis permutações dos caracteres. Porém, um texto de `20` caracteres geraria `20!` possíveis permutações. Se computássemos 100 milhões de valores por segundo, demoraríamos mais de 300 anos para computar todos. Logo, testar todas as possibilidades é inviável.

Como existem vários métodos diferentes de cifras de transposição, cada um necessita de uma abordagem diferente.

Nesses dois posts no StackExchange: [1] e [2], Ilmari Karonen mostra métodos para resolver manualmente cifras de transposição colunar.

Referências

Applied Cryptography, second edition

Como usar GPG

O software GPG (GNU's Privacy Guard) é, entre outras coisas, uma implementação do PGP (Pretty Good Privacy), que é um conjunto de especificações para comunicação encriptografada.

Instalando o GPGP

Se você está usando alguma distribuição Linux, BSD, Plan9, ou qualquer outro sistema operacional minimamente decente, é bem provável que o pacote `gnupg` está disponível no repositório de pacotes da sua distribuição. Portanto, basta instalar esse pacote usando seu gerenciador de pacotes. Caso você esteja usando aquelas distribuições sem gerenciadores de pacotes em que tudo deve ser instalado manualmente, então, você pode compilar o `gnupg` pelo código fonte.

Comandos básicos GPGP

Listar chaves disponíveis

Use o seguitne comando para listar as chaves disponíveis no seu computador

```
gpg --list-keys
```

Listar chaves segundo um padrão

Use o seguitne comando para para listar as chaves no seu computador segundo um padrão:

```
gpg --list-keys <pattern>
```

Por exemplo, para listar as chaves (armazenadas no computador) cujo nome do proprietário da chave ou email do proprietário contém o padrão Artix

```
gpg --list-keys Artix
```

Produz (no meu computador) a saída:

```
pub   rsa4096 2020-04-19 [SC] [expires: 2025-04-18]
      A574A1915CEDE31A3BFF5A68606520ACB886B428
uid           [ unknown] Christos Nouskas <nous@artixlinux.org>
```

Exportar sua chave pública para um arquivo

Use o seguinte comando para exportar sua chave pública para um arquivo:

```
gpg --export '<key>' > pubkey.gpg
```

Substitua `<key>` com o `id` da sua chave (pode ser o `Real name` associado com a chave, ou o `email` associado com a chave, ou o `id` da chave). Por exemplo:

```
gpg --export 'bruce@brucelee.com' > pubkey
```

Você pode querer exportar sua chave pública se você está distribuindo algum software (sua chave pode ser usada para verificar integridade e autenticidade do software) ou simplesmente para compartilhá-la com seus amigos para eles te enviarem mensagem encriptografadas.

Importar uma chave pública vinda de um arquivo

Use o seguinte comando para importar uma chave pública vinda de um arquivo:

```
gpg --import pubkey.gpg
```

Você pode querer importar a chave pública para verificar legitimidade de um software, ou importar a chave de seus amigos para enviar mensagem encriptografadas para eles.

Encriptografar um arquivo usando uma chave pública

Use um dos seguintes comandos (ambos são equivalentes) para encriptografar um arquivo usando uma chave pública:

```
gpg --encrypted --recipient '<key>' <file_path>
```

```
gpg -e -r '<key>' <file_path>
```

Por exemplo, para encriptografar um convite de festa que você enviará para o homem-aranha, use o comando:

```
gpg -e -r 'peterpark@spiderman.com' ~/documents/party-invitation.txt
```

Irá produzir o arquivo criptografado `party-invitation.txt.gpg`, que somente poderá ser decriptografado por quem deter a chave privada do homem-aranha.

Assinar um arquivo digitalmente

Use o seguinte comando para assinar um arquivo digitalmente:

```
gpg -b <file_path>
```

Isso irá gerar uma assinatura não-anexada, o que significa que o arquivo da assinatura está separado do arquivo original (a assinatura é um arquivo diferente).

Por padrão, o nome do arquivo da assinatura gerada será o nome do arquivo original concatenado com a string `.sig`.

Exemplo: Se o arquivo é `~/documents/letter.pdf`, a assinatura produzida estará localizada em `~/documents/letter.pdf.sig`.

É possível especificar o nome do arquivo gerado usando a flag `-o`, da seguinte forma:

```

```

```
gpg -b -o <output_file_path> <file_path>
```

Verificar a assinatura digital de um arquivo

Use o seguinte comando para verificar a assinatura digital de um arquivo:

```
gpg --verify <signature_file_path> <file_path>
```

Para isso funcionar, é necessário que você tenha, no seu computador, a chave pública usada para assinar o arquivo. Do contrário, não será possível verificar a assiantura digital.

A assiantura digital pode ser usada para verificar que um arquivo é legítimo, não foi corrompido, não foi alterado, e foi de fato assinado pelo dono da chave.

Exemplo: verificar a assinatura de um arquivo ISO:

```
gpg --verify manjaro-xfce-21.0.7-minimal-210614-linux510.iso.sig manjaro-xfce-21.0.7-minimal-210614-linux510.iso
```

Decriptografar um arquivo usando a chave privada

Somente é possível decriptografar um arquivo que foi encriptografado com a chave pública se você detiver a chave privada correspondente.

Use o seguinte comando para decriptografar um arquivo usando a chave privada:

```
gpg -d <file_path>
```

O conteúdo do arquivo será enviado para a saída padrão. Se você não deseja isso, redirecione a saída padrão para algum outro arquivo. Exemplo:

```
gpg -d mensagem.txt.gpg > mensagem.txt
```

Encriptografar um arquivo usando uma senha

Você pode encriptografar um arquivo usando uma senha, o que significa que você não precisa usar chave pública.

Use o seguinte comando para encriptografar um arquivo usando uma senha:

```
gpg -c <file_path>
```

Em seguida, você será perguntado para fornecer uma senha. Digite a senha e a confirmação da senha.

Decriptografar um arquivo usando uma senha

Use o seguinte comando para decriptografar um arquivo usando uma senha:

```
gpg -d <file_path>
```

Em seguida, você será perguntado para fornecer a senha. Digite a senha.

O conteúdo do arquivo será enviado para a saída padrão. Se você não deseja isso, redirecione a saída padrão para algum outro arquivo. Exemplo:

```
gpg -d mensagem.txt.gpg > mensagem.txt
```