Unip Chácara Santo Antônio-SP

Ciência Da Computação

Criptografia

Alunos - RA - Turma:

Antony Souza Da Cruz (F315GG6) (CC2A40)
Diego Reis de Magalhães (N596058) (CC2A40)
Kaique da Silva Ferreira (N671890) (CC2B40)
Matheus Ribeiro de Oliveira (N686EG5) (CC1A40)

São Paulo
2020

Unip Chácara Santo Antônio-SP

Alunos - RA - Turma:

Antony Souza Da Cruz (F315GG6) (CC2A40)
Diego Reis de Magalhães (N596058) (CC2A40)
Kaique da Silva Ferreira (N671890) (CC2B40)
Matheus Ribeiro de Oliveira (N686EG5) (CC1A40)

Criptografia

Trabalho Acadêmico apresentando a disciplina Introdução a Programação Estruturada Da Faculdade Unip, Campus Chácara Santo Antônio
Professor(a) Orientador(a): Myrian

São Paulo
2020

Índice

Objetivo do Trabalho

O objetivo do trabalho é desenvolver um programa que utilize a criptografia monoalfabética, restringindo o acesso de informações que serão trocadas entre inspetores especializados e autorizados pela guarda costeira durante a investigação da apreensão de um navio que transportava lixo tóxico da Ásia para a região norte do Brasil.

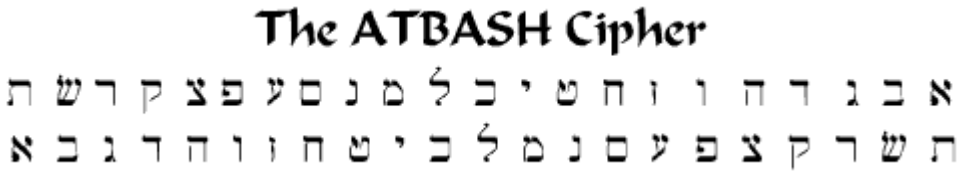
O método de criptografia utilizado será a Cifra de César, ela consiste na utilização de 26 possíveis chaves para a cifragem de um texto com no máximo 128 caracteres. Essa cifragem será feita através de um programa desenvolvido por meio da linguagem Python.

Para isso, quando o programa é iniciado, ele irá perguntar se você deseja criptografar ou descriptografar. Após isso, será solicitado a introdução de uma mensagem e uma chave numérica, que em seguida retornará já criptografada.

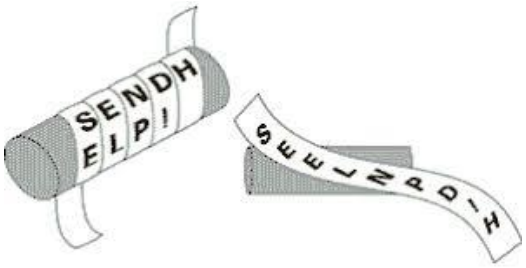
Introdução

A criptografia (em grego: kryptós, "escondido", e gráphein, "escrita"), surgiu para fins militares, sendo utilizada primeiramente pelos Hebreus com a necessidade de um meio de comunicação mais *segura*, dessa forma, escondendo informações do inimigo em meio a guerra por meio de códigos que apenas aliados conheciam.

O principal meio de criptografia utilizado pelos Hebreus era a "cifra de Atbash", ela consistia em uma cifra de substituição monoalfabética, onde cada letra do alfabeto era substituída por outra letra em um alfabeto invertido.

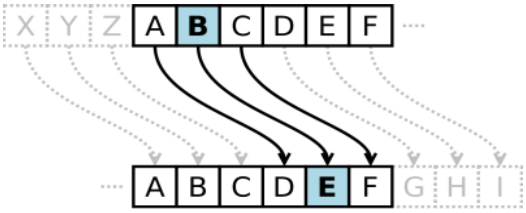


No mesmo contexto, tivemos o "Cítala" (Skytale), que era um método de criptografia espartana que consistia em enrolar uma tira de couro com uma mensagem embaralhada, escrita com várias outras letras aleatórias, em uma tora com o tamanho específico.



Dessa forma, os mensageiros enviados poderiam transportar as mensagens sem correr o risco do inimigo ter acesso às informações, pois mesmo que isso ocorra, ela não seria decifrada.

Aproximadamente 70 a.C. Júlio César criou a chamada "cifra de César", para ser utilizada em campos de batalhas como comunicação entre os seus generais. A cifra consiste em pular uma determinada quantidade de letras do alfabeto, tendo ao todo 26 possibilidades. Como por exemplo, se fosse pulado quatro casas, o A seria igual E.



Após aproximadamente 500 anos, a Cifra de César foi quebrada pelos árabes após identificarem que algumas letras se repetiam com frequência. Já a cifra de Atbash foi quebrada aproximadamente 300 anos depois, pelo matemático árabe Ibrahim Al-Kadi, que desenvolveu uma técnica para quebrar a cifra de Atbash.

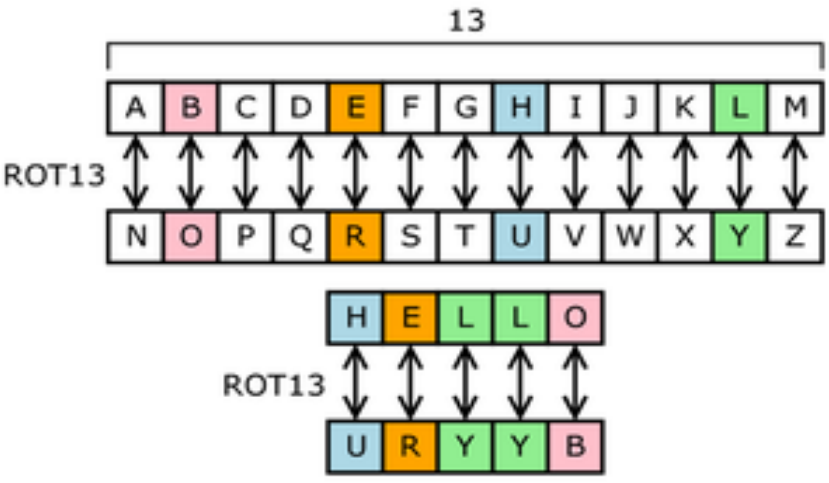
Por volta de 1465 d.C., Leon Battista Alberti desenvolve a “Cifra de Vigenère”, sendo ela uma cifra polialfabética que faz uso de diversas cifras de César. Porém, no mesmo século alguns criptoanalistas conseguiram quebrá-la.

Durante a Segunda Guerra Mundial, a área de criptoanálise teve grandes avanços, sendo um deles as máquinas denominadas como Enigma, desenvolvida pelos alemães e patenteada por Arthur Scherbius em 1918 d.C. Seu funcionamento tem como base o aprimoramento da cifra de Vigenère e consistia em sua engenhosa engrenagem, que tornava a criptografia quase impossível de ser decifrada por humanos, ela era capaz de gerar milhares de combinações para ser aplicada na criptografia de texto.

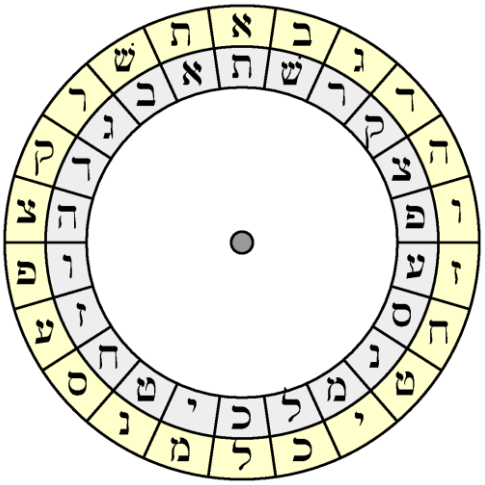
Criptografia (Conceitos Gerais)

A criptografia consiste, de maneira geral, em esconder informações contidas no texto, onde somente o remetente e destinatário tenham acesso ao conteúdo escondido. O primeiro relato de aplicação da criptografia, foi a cifra de transposição (*Cítala* ou *Skytale*) utilizada para fins militares e criada pelos Espartanos, ela utilizava uma técnica de embaralhamento de letras contidas em uma mensagem, e apenas a pessoa que possuía o objeto exato, poderia reorganizar-lá. A mensagem escrita era embaralhada em uma tira de couro e, somente desembaralhada, caso fosse enrolada ao redor de um bastão de madeira com um tamanho e forma específicos (se as dimensões do bastão fossem diferentes, o couro não se alinharia e não revelaria a mensagem).

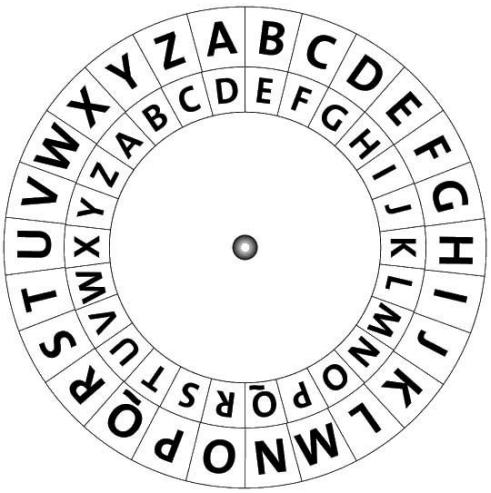
No mesmo período tivemos a “cifra de Atbash” que fazia uso de uma criptografia de embaralhamento das letras, sendo ela uma cifra monoalfabética. A criptografia monoalfabética funciona da seguinte forma: possibilita, qualquer letra do alfabeto pode ser substituída por outra, contanto de que cada letra seja substituída por uma única letra.



A cifra de Atbash é uma criptografia antiga, desenvolvida pelos Hebreus para esconder suas mensagens de guerra. Essa cifra consistia em trocar todas as letras do alfabeto hebraico pelo mesmo alfabeto, porém escrito da forma inversa. Dessa forma “א” era trocado por “ת“ ,”ש" por “ב” e assim por diante.



Após vários anos, outra cifra monoalfabética surgiu na Grécia Antiga, conhecida como “Cifra de César”. A cifra de César, também criada com intuito militar, seguia um padrão regular de embaralhamento de letras, nela era utilizado uma substituição por deslocamento. Como por exemplo, caso o deslocamento fosse de 3 letras, sendo a letra “A” igual a um somado com mais três. Desta forma a letra “A” se tornaria “D”, “B” se tornaria “E” e assim por diante, sendo a chave S=3.



Mesmo que a Cifra de Atbash e a Cifra de César sigam um padrão de embaralhamento, isso não seria necessário para serem consideradas monoalfabéticas. Na cifra monoalfabética as letras podem ser trocadas por qualquer outra letra, sendo a única limitação que cada uma seja substituída por uma única letra, ou seja, temos 26 pares de letras possíveis, sendo ela muito melhor que as cifras de Atbash (somente 1 par possível) e a de César (somente 26 pares possíveis).

Com o passar dos anos, a criptoanálise foi se aperfeiçoando, e, dessa forma sendo possível, por meio da lógica, quebrar as cifras monoalfabéticas. Criptoanalistas árabes perceberam que textos cifrados continham uma utilização mais frequente de algumas letras, sendo assim, quebrando a criptografia com mais facilidade.

Por exemplo, se o criptoanalista tiver conhecimento que o texto original foi escrito em inglês, ele poderá deduzir que as duas letras mais frequentes do texto cifrado sejam "e" e "t" (as duas letras mais frequentes em textos em inglês), e que conjunto de letras mais frequente sejam "in", "it", "the", "ion" e "ing" (pois também se repetem com grande frequência).

Além da frequência de letras, também podem ser previamente analisadas algumas palavras chaves caso a pessoa que está analisando a cifra tiver algum conhecimento prévio do conteúdo do texto original. Por exemplo, se o texto está sendo designado a uma pessoa, provavelmente o nome desta pessoa estará presente no texto.

Assim, somente com o conhecimento da língua utilizada e a quem o texto se designa, muitas possibilidades podem ser descartadas, desta forma, facilitando a decifragem do texto se ela estiver sendo feita por meio de força bruta, sendo esse um dos pontos negativos da utilização de uma cifra monoalfabética.

Além disso, como citado no livro "Redes de Computadores e a Internet" de Kurose Ross, uma análise à força bruta pode ser dividida em três diferentes cenários, sendo eles os seguintes:

- Ataque exclusivo a texto cifrado - Quando o intruso pode ter acesso somente ao texto cifrado interceptado, sem ter nenhuma informação exata sobre o conteúdo do texto aberto.

- Ataque com texto aberto conhecido - Quando o intruso tem algum conhecimento prévio do texto, como língua escrita e palavras que possam estar presentes no texto.

- Ataque com texto aberto escolhido - Quando o intruso pode escolher a mensagem em texto aberto e obter seu texto cifrado correspondente.

Devido à quebra de mensagens cifradas com a cifra monoalfabética, uma nova técnica foi desenvolvida, sendo ela a criptografia polialfabética. Diferente da monoalfabética, esta nova técnica não se limitava a substituição de uma letra somente por outra, agora uma letra podia ser representada por diversas letras em um único texto.

Essa técnica de criptografia utiliza várias cifras monoalfabéticas para realizar seu embaralhamento, fazendo uso de uma cifra monoalfabética por cada letra para codificar sua mensagem.

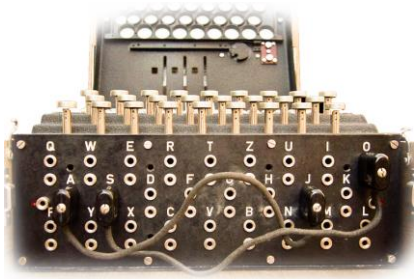
Um exemplo de cifra polialfabética é a “Cifra de Vigènere”, que fazia uso de um mapa com diversas letras embaralhadas. Nessa cifra, primeiramente era escolhida a chave, ela poderia ser uma palavra, frase ou até mesmo um texto. Após a chave ser escolhida, será utilizado o mapa para codificar sua mensagem. Já no mapa, cada letra da chave e do texto aberto era utilizado para encontrar no mapa qual letra será utilizada para substituir a letra do texto aberto.

Por exemplo, se tivermos como chave a palavra “chave” e como texto aberto a frase “texto para codificar”, será utilizado a letra “c” e a letra “t” para encontrar no mapa qual letra irá substituir a letra “t” no texto aberto. Sendo o texto codificado como “vlxos rhrv gqkiamehr”.

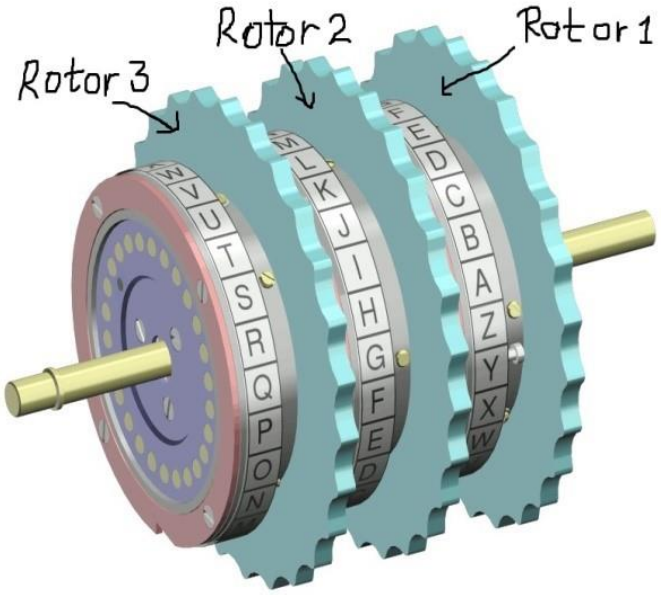
Por ser uma cifra bem mais complexa que a cifra de César, foram necessários cerca de 300 anos para quebrar a cifra de Vigènere. Kasiski decifrou e publicou seu método de decodificação, sem a necessidade de saber qual era a chave da mensagem.

O Método de Kasiski funciona da seguinte forma, suponha-mos que a frase “kssij ck qkfycjiy” fosse interceptada no meio de seu envio por um especialista em segurança, para decodificar, é preciso encontrar uma falha na cifra, essa falha se encontra na chave, e para encontrar a chave é preciso conhecer a sequência de repetições dela, para isso basta saber quantas vezes uma letra do alfabeto se repete e o intervalo de tempo em que isso acontece, tendo isso em mente, é possível analisar qual das letras mais se repete, linha por linha, a letra que mais aparece, é a que faz parte da chave (esse modo se chama análise de frequência). No final da análise a mensagem “kssij ck qkfycjiy” será lida “somos os melhores”, e sua senha “SEGURO”.

Anos depois, na 2°Guerra Mundial, surgiu um novo tipo de criptografia que se destacou por sua extrema complexidade, se chamava “Enigma”.Trata-se de uma máquina que é “uma mistura” da cifra de César, Vigènere e de chaves rotativas.



Desenvolvida pelos nazistas e utilizada pelos Alemães, o “Enigma” é muito parecida com uma máquina de escrever antiga, mas sua construção implementa três rotores e um sub-rotor. Ela era utilizada para enviar mensagens em canal aberto (frequências de rádio), para bases aliadas durante a guerra. Ela era tão segura que mesmo se os inimigos conseguissem se conectar na mesma frequência e captar essa mensagem, não iriam desvendá-la, pois a chave possuía milhões de possibilidades e a letra nunca se repetia.



Seu funcionamento se baseia na rotação dos três rotores combinados, na qual, cada um terá uma letra da chave, e conforme a mensagem é escrita, os rotores giram de formas distintas alterando a saída sem repetir uma mesma letra de saída para uma de entrada, por exemplo, se fosse digitado “AAA” a cifra Nunca sairia “EEE”, mas sim “FSY”, ou seja, as letras que eram criptografadas nunca se repetiam, e isso era o ‘que tornava a máquina indecifrável. Sua chave era programada através dos ponteiros dos rotores, e era criada no mesmo momento em que os ponteiros se movimentavam, isso significa que para obter a chave era preciso saber a posição dos rotores.

O responsável por desvendar a máquina "Enigma" foi o matemático Alan Turing, mais conhecido pelo filme “Jogo da Imitação”.

Alan Turing conseguiu desenvolver uma máquina para testar chaves possíveis nas cifras interceptadas das transmissões de rádio. Inicialmente, a máquina de Alan foi programada para quebrar em todas as 17.576 chaves possíveis (26^3), mas o grande problema era que a cada 24 horas as chaves eram trocadas pelos alemães, isso tornava impossível desvendar a cifra e verificar as mensagens a tempo.

Segundo fontes não conhecidas, Alan conheceu uma mulher de sua equipe que era apaixonada por um soldado alemão, e por conta disso, conseguiu descobrir que as mensagens sempre continham a frase “Heil Hitler” que é na verdade uma saudação a Hitler. Seguindo sua incrível lógica, Alan pensou que, “Heil Hitler” pode ser codificada com 17.576 chaves possíveis, antes da próxima mensagem ser captada, portanto, em todas as mensagens ele sempre procurava “Heil Hitler” no início ou no fim da mensagem, e dessa forma é possível descobrir a chave da cifra e assim descriptografar e descobrir a mensagem.

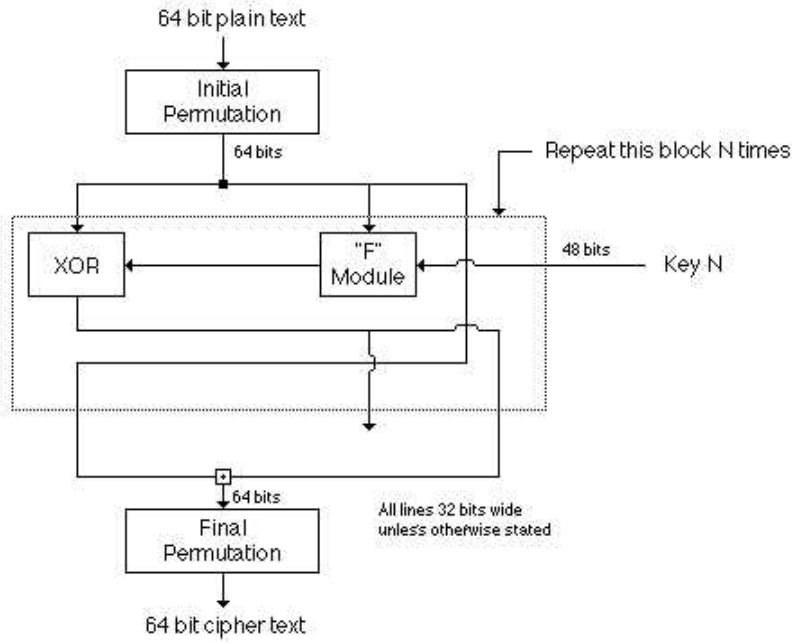
Muitas cifras foram criadas desde o enigma, sendo elas bem mais complexas que tais, onde na maioria fazendo uso de hex ou binário, com uma quantidade de possíveis combinações beirando o infinito, onde um computador quântico levaria anos para quebrá-las. Desta forma sendo utilizados os “bits” são utilizados para denominar a quantidade de suas possíveis combinações, sendo utilizados até 256 bits nos meios de criptografia mais avançados.

Além disso, hoje em dia a criptografia deixou de ser utilizada exclusivamente para fins militares e passou a ser utilizada como um meio de segurança em geral, mas mantendo em sua essência de esconder algo. Como por exemplo, sendo utilizada para manter privadas as suas informações armazenadas em servidores de bancos, lojas de comércio eletrônico etc.

Criptografias mais utilizadas

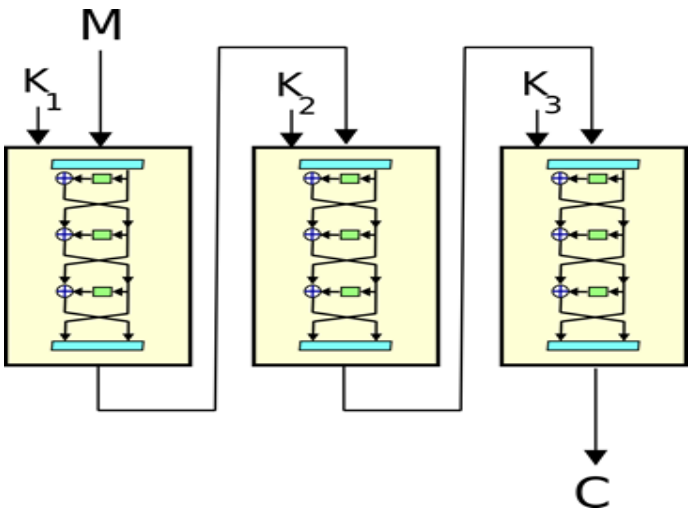
A criptografia DES (Data Encryption Stardard) é um algoritmo de criptografia que foi criado na década de 70 pelo *National Bureau of Standarts*. Teve como objetivo criar um padrão para proteção de dados, o primeiro rascunho do algoritmo foi criado pela IBM, inicialmente chamado de Lúcifer.

Nesta mesma época em que foi criada ela foi utilizada em grande escala internacional, após ser estudado, o DES foi modelo de inspiração para inúmeros modelos de criptoanálise muito utilizados nos dias de hoje. O DES não é considerado muito seguro nos dias de hoje, por conta de testes e estudos teóricos que foram realizados, comprovando suas falhas perante sistemas mais complexos.



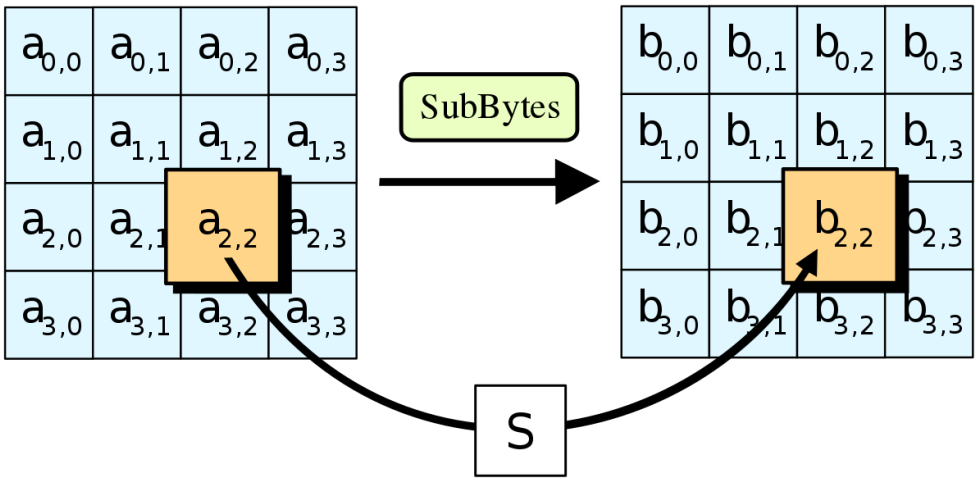
O 3DES (Triple Data Encryption Standard) foi desenvolvido pela IBM em 1974, tem sua criptografia simétrica baseada no algoritmo do DES, porém utiliza 3 chaves de 64 bits, sendo 8 bits de cada chave para verificar paridade, sendo assim dando um tamanho total de 168 bits para a chave, trata-se de uma cifra simétrica, a decriptação é realizada por meio das mesmas chaves utilizadas para a encriptação, os dados são encriptados com a primeira chave, decriptado com a segunda chave e finalmente encriptado novamente pela terceira chave. Entretanto, para a

decriptação, a ordem das rotinas é invertida, isto faz com que o 3DES seja mais lento que o DES original, porém muito mais efetivo em segurança.

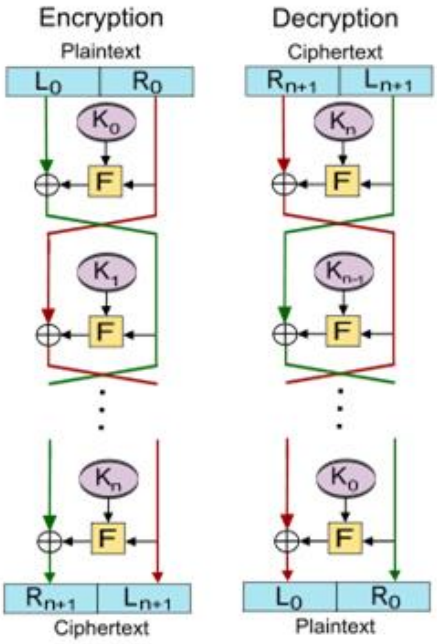


AES (Advanced Encryption Standard), criptografia de dados eletrônicos estabelecida pelo Instituto Nacional de Padrões e Tecnologia (NIST) dos EUA em 2001. É possivelmente umas das criptografias simétricas mais utilizadas no mundo, também por conta de ser compatível com inúmeros sistemas operacionais, e foi criada com o objetivo de ser mais rápida do que o antigo padrão do 3DES. Seu funcionamento é baseado em três chaves com um tamanho de bloco de 128 bits, porém as chaves têm tamanhos diferentes: 128, 192 e 256 bits.

No AES depende do tamanho das chaves, o algoritmo possui uma chave principal, e a partir dela são geradas outras chaves chamadas de “chaves de rodada”, pois cada uma é usada em rodadas diferentes.



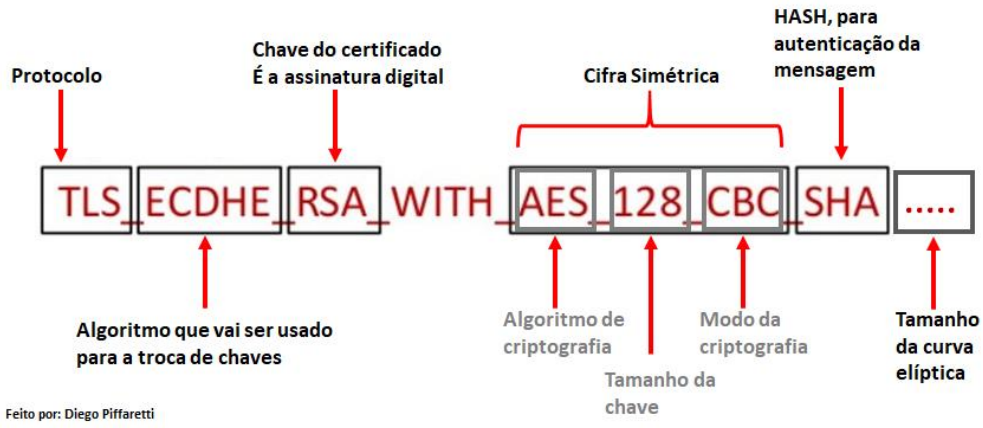
Data Encryption Standard-X criptografia simétrica de bloco que também foi baseada na DES, porém utilizando uma técnica conhecida como *key whitening*, que visa dificultar ataques de força bruta sem que seja necessária uma complexidade computacional. Seu funcionamento se baseia em adicionar 64 bits na encriptação, aumentando a proteção contra força bruta. Atualmente essa encriptação pode ser quebrada por ataques mais sofisticados (é um programa que evolui a cada tentativa de decifração).



Criptografia Camellia inicialmente foi desenvolvida pela Mitsubishi Electric do Japão. Sua segurança e processamento é comparado à *Advanced Encryption Standard* (AES). Foi projetada com criptografia que fornece segurança das comunicações com implementações para software e hardware.

Seu funcionamento constitui em chaves de 128, 192 e 256 bits, que sofrem uma transformação chamada de “FL-função”.

Camellia é considerada uma cifra segura e moderna. Mesmo utilizando um tamanho menor de chave (128 bits). Não há ataques bem sucedidos conhecidos à esta cifra.



A criptografia RSA (Esse nome indica a primeira letra do sobrenome de cada criador) foi criada em 1977 por 3 cientistas que trabalhavam no MIT: Ron Rivest, Adi Shamir e Leonard Adleman. Ela foi a primeira “criptografia assimétrica” criada até então, trouxe inovação por utilizar 2 chaves (uma pública e outra privada) durante o processo de criptografar e descriptografar uma informação. De forma resumida, a chave pública criptografia e a chave privada descriptografa. Essa criptografia é considerada mais eficiente e segura do que a simétrica, que é conhecida por utilizar apenas 1 chave pública durante todo o processo.

RSA foi um grande avanço para a época e é considerada, até hoje, uma das melhores e mais seguras criptografias já criadas de todos os tempos. Com 1 chave, era necessário enviar a mesma chave do emissor para o receptor, podendo ser facilmente interceptada por terceiros durante esse processo. Agora, todos os usuários já possuem uma chave privada especifica que somente elas têm acesso, então apenas quem possuir essa chave irá conseguir descriptografar a mensagem/informação.

Na imagem abaixo é mostrado o processo detalhado dessa criptografia, mas, resumidamente, utiliza-se de 2 números primos gigantescos que irão passar por inúmeros processos, para descobrir os valores iniciais é preciso fatorar inúmeras vezes até chegar no valor inicial (esses valores seriam a "quebra" da chave, a senha), no entanto esse processo é bem complexo e pode levar até anos para ser decifrado manualmente.

No RSA as chaves são geradas da seguinte maneira:

1. Escolha de forma aleatória dois números primos grandes ***p*** e ***q***, da ordem de 10^{100} no mínimo
2. Calcule ***n=pq***
3. Calcule a função totiente* em ***n: $\phi(n) = (p - 1)(q - 1)$***
4. Escolha um inteiro ***e*** tal que ***$1 < e < \phi(n)$***, de forma que ***e*** e ***$\phi(n)$*** sejam co-primos (ou seja, o único divisor comum seja 1)
5. 5 - Calcule ***d*** de forma que ***$de \equiv 1 \pmod{\phi(n)}$***, ou seja, ***d*** seja o inverso multiplicativo de ***e*** em ***$\pmod{\phi(n)}$***.

Por fim, temos: A chave pública: o par ***(n, e,)***, e a chave privada: a tripla ***(p, q, d)***.

A criptografia IDEA (International Data Encryption Algorithm) foi criada em 1991 na Suíça por James Massey e Xueija Lai. Ele é criptografia simétrica (ou chave clássica), ou seja, utiliza-se de apenas uma chave pública (128 bits) para criptografar e descriptografar informações, também é classificado com cifra de blocos (agrupamento de bits de tamanho fixo, 64 bits no caso) . Basicamente a IDEA é uma DES só que mais rápida, tanto que foi criada com esse objetivo. Hoje em dia ela é amplamente utilizada em diversas áreas, com uma comunicação segura até internet banking.

IDEA possui 4 modos de operação: ECB, CBC, CFB e OFB, sendo ECB considerado o menos eficaz. IDEA envolve muitos processos matemáticos durante o seu processo de criptografar, ele utiliza repetidamente as operações: adição, multiplicação e XOR. As imagens abaixo ilustram o processo detalhadamente.

a) XOR bit-a-bit;

b) adição de inteiros módulo 2^{16}, com entradas e saídas tratadas como inteiros de 16 bits sem sinal, isto é, variáveis com valores somente positivos (*unsigned*);

c) multiplicação de inteiros módulo $2^{16} + 1$, com entradas e saídas tratadas como inteiros de 16 bits sem sinal, exceto o bloco constituído por zeros que é tratado como representação 2^{16}.

Estrutura do IDEA:

a) o texto claro de 64 bits é dividido em quatro partes;
b) a partir da chave de 128 bits são geradas 52 sub-chaves de 16 bits cada;
c) o IDEA é composto por oito rodadas, sendo que em cada rodada são aplicadas 6 sub-chaves, e a mesma sub-chave não é repetida;
d) ao final, os dados passam por uma transformação.

A criptografia *Blowfish* foi criada por *Bruce Schneier* em 1994, classificado com cifra de blocos (64 bits). Assim como o IDEA e a DES, ele é uma criptografia simétrica, ou seja, utiliza-se apenas uma chave pública. Um diferencial desse algoritmo é que ele é GRATUITO e NÃO PATENTEADO, considerado um dos melhores algoritmos de criptografia gratuitos disponíveis. Sua chave pode variar entre 32 a 448 bits, uma de suas desvantagens é a necessidade de memória relativamente grande se comparada com outros algoritmos do mercado.

Blowfish é utilizado em alguns sistemas operacionais, como o Linux, também pode ser utilizado em: gerenciamento de senhas, e-commerce e banco de dados.

De forma leiga, O processo do *Blowfish* é bem semelhante ao IDEA, utiliza-se de 3 operações matemáticas (adição, multiplicação e XOR) e na transformação da chave em sub-chave, fazendo aumentar o total de bits durante a operação. Uma parte do processo está na imagem abaixo:

A entrada para essa parte do algoritmo são 64 bits, que serão divididos em dois grupos de 32 bits, que serão chamados de xL e xR. As operações abaixo deverão ser feitas 16 vezes.

- xL = xL XOR Pi
- xR = F(xL) XOR xR
- Troca de xL com xR

Após a décima sexta iteração, é necessário trocar xL e xR mais uma vez (*troca de xL e xR*). Em seguida, são feitas as seguintes operações:

- R = xR XOR P17;
- xL = xL XOR P18.

O texto cifrado será a união desses dois grupos (xLxR). A função F segue os seguintes passos:

A criptografia *Twofish* foi sucessora do *Blowfish*, criada em 1998 também por *Bruce Schneier*. Esse algoritmo foi um dos cinco finalistas no concurso *Advanced Encryption Standard* (AES), infelizmente não chegou a ganhar. Assim como o *Blowfish*, ele também é um algoritmo de cifra de blocos (128 bits, diferente do seu antecessor que tinha 64 bits), criptografia simétrica com chave pública que varia entre 3 valores: 128,198 ou 256 bits. Assim como seu antecessor, *Twofish* também é gratuito e não tem patente. Processo semelhante ao do *Blowfish.*

O *Twofish* proporcionou uma maior segurança se comparado com o *Blowfish,* isso se dá graças ao grande aumento de bits na cifra de blocos.

A criptografia SAFER foi criada em xxxx por *James Massey* (um dos criadores da IDEIA), é classificado como criptografia simétrica (apenas 1 chave pública) e também é uma cifra de blocos, algoritmo 100% focado na segurança. Existem várias versões desse algoritmo que variam a quantidade de bits, a seguir as versões existentes.

- Safer: 64 bits
- Safer SK: 64 bits (mais segurança)
- Safer-128: 128 bits
- Safer SK-128: 128 bits (mais segurança)

Resumidamente, o processo do algoritmo consiste em dividir a chave em várias sub-chaves e, logo em seguida, realizar 8 passos de operações (soma e XOR). Abaixo estão imagens da ordem dos 8 passos e um exemplo de sub-chave.

```
XOR, add, add, XOR, XOR, add, add, XOR

  1  45 226 147 190  69  21 174
120   3 135 164 184  56 207  63
  8 103   9 148 235  38 168 107
189  24  52  27 187 191 114 247
```

.

Dissertação (Cifra de César e sua evolução)

A cifra de César é uma das técnicas mais conhecidas e simples de criptografia. Inicialmente, ela foi utilizada por Júlio César em torno de 58 a.C., na qual ele mudava cada letra de seus comandos militares fazendo parecer sem significado, caso essa mensagem chegasse nas mãos dos inimigos. Líderes militares ainda utilizaram essa técnica por centenas de anos depois de César, mas depois de 800 anos, um árabe matemático descobriu e publicou o funcionamento da cifra, o ‘que a tornou defasada, em relação à segurança de suas informações, assim como afirma *Brit Cruise* (especialista de segurança da informação-Khan Academy), que diz “ a análise de frequência foi um golpe na segurança da cifra de César”.

A cifra de César trata-se de um tipo de cifra de substituição, onde cada letra de um texto é substituída por outra letra no alfabeto deslocando-se um certo número de posições para a esquerda ou para a direita. Um exemplo seria se utilizássemos uma troca de três posições para a direita, sendo assim, cada letra do texto é substituída pela letra que fica três posições adiante, nesse caso a letra A seria substituída pela letra D, a letra B pela letra E e a letra C substituída pela letra F.

As técnicas que eram utilizadas anteriormente à Cifra de César, se baseavam na substituição de letras do alfabeto (idioma escolhido) para criptografar mensagens, sendo elas a Cifra de Atbash e a Cifra de Cítala.

A cifra de Atbash, é uma cifra hebraica e umas das mais conhecidas, baseia-se em um sistema de substituição monoalfabética utilizando a inversão do alfabeto como base de criptografia. Dessa maneira seria estruturada da seguinte forma: a letra A que é a primeira letra do alfabeto seria substituída pela última letra do alfabeto (Z), a letra B que é a segunda letra seria substituída pela penúltima letra (Y), seguindo assim sucessivamente.

A cifra de Cítala é um sistema de criptografia utilizado pelos gregos para enviar mensagens sem que o inimigo pudesse decifrá-las, é formada por duas varas e uma tira de couro ou papiro. A mensagem era escrita longitudinalmente, de forma

que cada volta da tira aparecesse uma letra, o receptor só tinha que enrolar a tira em sua vara para receber a mensagem original.

Desta forma, a Cifra de César se destacava em relação aos anteriores por seu nível de complexidade e praticidade. Diferente da Cítala, a cifra de César seguia uma ordem pré-determinada (chave) para a posição de cada letra que seria criptografada, sendo possível criptografar textos sem qualquer limitação de tamanho. Já em Cítala, sua criptografia acabava sendo mais trabalhosa de se fazer, sendo necessário uma tora de espessura específica para cada palavra.

Na Cifra de Atbash, a ordem de posição para cada letra era apenas uma, que consistia em reverter a sequência alfabética. Enquanto na de César, graças a criação das chaves numéricas, existem 26 possibilidades para se seguir, sendo uma utilização mais complexa do sistema monoalfabético.

Porém, mesmo sendo ela muito eficaz em sua função em relação com as técnicas anteriores, acabava por ter diversas vulnerabilidades que fizeram com que seu código fosse posteriormente quebrado, sendo necessário criar novos métodos mais eficazes de criptografia, até chegarmos nos dias atuais.

O principal problema da cifra de César é seu padrão no embaralhamento de letras, onde há somente 26 possibilidades de embaralhamento, sendo uma dessas possibilidades o próprio texto original. Isso se dá ao fato de que, por ser uma cifra de substituição monoalfabética, cada letra poderá ser somente substituída por uma letra, e que todas devem seguir a sequência do alfabeto, iniciando pela letra escolhida pela chave.

Alfabeto Normal:

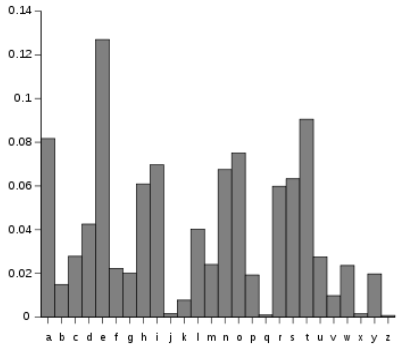
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z

Alfabeto Cifrado:

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

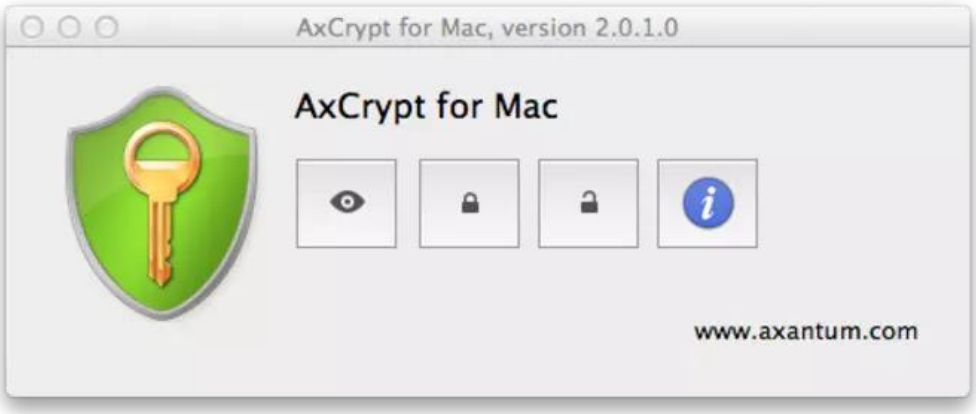
Desta forma, um invasor conseguiria descobrir a mensagem por meio da força bruta em poucos minutos sem muita dificuldade, somente testando cada possibilidade ao saber que o texto foi cifrado por meio de César.

Além da falta de possíveis combinações de embaralhamento, também é possível descobrir a mensagem através da análise de frequência. Como por exemplo, na língua portuguesa a letra “e” é a que mais se repete, logo a letra que mais se repetir no texto cifrado poderia ser substituída pela letra “e” e dessa forma diminuindo o número de possibilidades para uma.



Sendo assim, embora a cifra de César tenha conseguido permanecer intacta por muitos anos, é um método de cifragem muito arcaico e não poderia ser utilizado nos dias de hoje, pois mesmo que à primeira vista pareça complexo, acaba sendo uma cifra relativamente simples e fácil de ser decodificada. E por esta razão, passou a ser uma, ou senão a cifra mais utilizada para estudos e introdução à criptografia. Porém, todo grande e bom projeto, se compõem por diversas técnicas e não apenas uma. Nesses casos a cifra de César tem grande utilidade.

Existem ferramentas que ainda utilizam a cifra de César e visam a segurança da mensagem ou do usuário (varia da aplicação e do contexto em que vão ser utilizadas), como por exemplo o software *Axantum*, é uma ferramenta poderosa para criptografar mensagens e dados, que foi criada em 2016 pela empresa *Axantum Software AB*(2004) e depois transferido para uma empresa separada, *AxCrypt AB*(2016).



Também existem ferramentas que utilizam do conceito da cifra de César para enviar e receber dados, como o *CyberGhostVPN*, é um software que utiliza a VPN (Rede Virtual Privada) para enviar dados Criptografados, essa ferramenta foi criada em 2005 pela empresa *CyberGhost.*

A cifra de César teve grande importância na sua época para fins militares, e atualmente ela é a principal técnica utilizada para lesionar aprendizes na área de criptografia. Fazendo uma comparação no cotidiano, a cifra de César não é tão utilizada por ser mais simples, quando comparada com outras técnicas mais atuais, porém ela é muito importante para estudos sobre criptografia.

Se comparada com a de Viginére, a cifra de César é bem mais ultrapassada, pelo seu déficit de possíveis combinações, tornando-a fácil de ser quebrada por força bruta ou análise de frequência. Enquanto a de Viginére utiliza-se de uma

tabela para realizar sua codificação, dessa forma podendo ter uma grande quantidade de combinações e dificultando a sua decodificação.

Porém, a Cifra de Vigenere também teve sua segurança quebrada através da análise de frequência, o que a tornou inviável em questões de segurança, sendo utilizada somente para estudos. Anos depois, os Nazistas criaram a máquina Enigma, que utilizava a técnica de Vigenere de uma forma aprimorada, consistia em gerar as chaves no mesmo momento em que as letras do texto eram codificadas, isto tornava a segurança da máquina humanamente inquebrável.

Desta forma, a máquina Enigma era muito mais desenvolvida que a cifra de Vigenere, tendo ela ainda mais possibilidades de embaralhamento. E, em relação a cifra de César, ela era infinitamente mais complexa, contendo 17.576 combinações possíveis, enquanto a de César possuía apenas 26 chaves.

Mesmo sendo ultrapassada, a cifra de César teve melhorias enquanto era considerada atual, que foi a criação da "Cifra monoalfabética", nela consiste uma chave diferente para cada letra da mensagem que será criptografada (existem 26 chaves, cada uma indica uma letra do alfabeto diferente para ser o A) , ou seja, cada letra tem uma ordem do alfabeto diferente. Essa técnica deixa a cifra de César muito mais complicada de ser decodificada por terceiros.

Texto Original	ABCDEFGHIJKLMNOPQRSTUVWXYZ
Texto Cifrado	UMGSCWXZFPBNTKHEYOVRDIAJLQ

Na imagem acima é possível perceber que, mesmo o texto original sendo o alfabeto na ordem certa, o texto cifrado não tem nenhum padrão perceptível. Cada letra tem uma chave diferente.

Também tem a versão mais simplificada da Cifra monoalfabética apresentada acima, ela tem o mesmo intuito, criar diferentes tipos de chaves para não deixar a Cifra de Cesar original tão previsível (Na original, o A sempre é equivalente ao D). Porém, nessa cifra simplificada é utilizada apenas 1 chave para a mensagem toda, ou seja, enquanto a cifra monoalfabética original usa 1 chave diferente para cada letra, essa mais simplificada usa apenas 1 chave para a mensagem inteira. No nosso trabalho, foi utilizado essa técnica mais simplificada da cifra de César monoalfabética.

Com isso, podemos notar a evolução e complexidade da cifra de César durante todos esses anos desde sua criação. Apesar de no começo ser bem básica e fácil de entender (foi criada em uma época bem remota), com o passar do tempo, adquiriu sua própria identidade e dificuldade. Até hoje é considerada uma das mais importantes por ser uma das pioneiras nesse mundo de criptografia.

Projeto

O funcionamento do programa consiste praticamente em 4 funções principais sendo elas:

Modo: Cria uma string ou letra de entrada para a seleção da função de encriptação da mensagem ou de decriptografia da mesma

Mensagem: Pede ao usuário a mensagem que será criptografada ou descriptografada.

Chave: Pede ao usuário uma chave numérica de 1 a 26.

Mensagem Traduzida: Função responsável pela encriptação e desencriptação da mensagem, sendo necessário o modo, mensagem e chave para seu funcionamento.

Desta forma, o programa funcionará do seguinte modo:

- Importação das bibliotecas “sys“e ”os”
- Criação da variável numérica “Tamanho”, que recebe 26;
- Exibirá “Pressione ENTER para começar” para o usuário;
- Criação da função “reiniciar”
  - Criação da variável “python” e atribuição da variável “executable” importada da biblioteca “sys” para ela;
  - Utiliza-se a função “execl” importada da biblioteca “os”;
- Criação da função “Modo”;
  - Quando o loop é verdadeiro realiza as seguintes linhas:
    - criação da variável de string Modo que solicitará que o usuário introduzido “e” ou “d”;
      - caso ele digite “e” ou “d” ela irá retornar modo;
      - senão irá pedir para que ele insira novamente;
- Criação da função “Mensagem”
  - Exibe “Digite a Mensagem” para o usuário e retorna o valor de entrada para o programa.
- Criação da função “Chave”
  - A variável “chave” recebe o valor “0”

        - É criado um “while”, uma estrutura de repetição que tem como condição “True”, ou seja, irá rodar até um comando cancelar o looping (no caso do nosso código, seria o “return”)
        - Exibe “Digite uma chave numérica (1-%s)' % (Tamanho))” para o usuário e retorna o valor de digitado para a variável “chave”.
        - Estrutura “if” que verifica se a variável “chave” está entre 1 e 26 (26 é o valor da variável “Tamanho”);
            - Se a condição for verdadeira, retorna a variável “chave”;
    - Criação da função “Mensagem_Traduzida”
        - Estrutura “if”, que irá verificar a primeira letra de “Modo”;
            - caso for “d” a variável “chave” recebe seu valor invertido;
        - A variável “traducao” é criada e recebe o valor “” (vazio);
        - Estrutura “For” de símbolo na Mensagem, que irá fazer a seguinte operação para cada caractere da mensagem;
            - Estrutura “if” que irá verificar se esse caráter pertence ao alfabeto;
            - A variável “num” recebe o valor ord da variável "símbolo" (ord é o comando de transformar uma letra em um número correspondente na cifra de César)
            - A variável “num” recebe o valor resultante da soma dela mesma com a variável “chave”
            - Estrutura “if”, que irá verificar se a letra está em maiúsculo;
                - Estrutura “if”, que irá verificar se o ord da letra é maior que o ord de Z (26)
                    - caso verdadeiro irá realizar 26 subtraído por “num”
                - Estrutura “elif”, que irá verificar se o ord da letra (variável “num”) é menor do que o ord de A (1)
                    - caso verdadeiro, irá realizar 26 somado por “num” e seu resultado será atribuído a “num”

- Estrutura “if”, que irá verificar se a letra está em minúsculo;
    - Estrutura “if”, que irá verificar se o ord da letra é maior que o ord de z (26)
        - caso verdadeiro irá realizar 26 subtraído por “num”
    - Estrutura “elif”, que irá verificar se o ord da letra (variável “num”) é menor do que o ord de a (1)
        - caso verdadeiro, irá realizar 26 somado por “num” e seu resultado será atribuído a “num”
- Estrutura “else” que realizará a seguinte operação:
    - Para a variável “traducao” será atribuído a soma de si própria com a variável "símbolo"
- Será retornado a variável “traducao”

- Declaração da variável “Modo” que receberá o valor retornado pela função “Modo”
- Criação da variável “Loop” e atribuição do valor lógico “True” a ela;
- Utilização de um “while” que estará ativo enquanto a variável “Loop” for verdadeira;
    - Declaração da variável “Mensagem” que receberá o valor retornado pela função “Mensagem”;
    - Criação da variável “total” que recebe o valor “0”;
    - Estrutura “for” com a criação da variável “letras”, que irá repetir de acordo com o tamanho de caracteres da variável “mensagem”;
        - Para cada caractere da variável mensagem será atribuído total + 1;
    - Estrutura “if” que executará caso a variável “total” seja maior do que 128;
        - Exibirá “----------”
        - Exibirá "Limite de 128 caracteres excedido"
        - Exibirá “----------”

            - Declaração da variável “reiniciar” que receberá o valor retornado pela função “reiniciar”
        - Estrutura “else” que será executada caso o “if“a cima seja falso;
            - A variável “Loop” receberá o valor lógico “False”;
    - Declaração da variável “chave” que receberá o valor retornado pela função “Chave”;
    - Será exibido para o usuário “Seu texto é:”;
    - Será exibido a variável “traducao” retornada pela função “Mensagem_Traduzida”, utilizando as variáveis “modo, mensagem, chave”;
    - Será exibido "----------" para o usuário;
    - Declaração da variável “reiniciar” que receberá o valor retornado pela função “reiniciar”.

Relatório com as linhas de código

Criptografia de César em Python

Unip Chácara Santo Antônio- Santo Amaro

Curso: Ciência da Computação

Alunos - RA - Turma:

Antony Souza Da Cruz (F315GG6) (CC2A40)
Diego Reis de Magalhães (N596058) (CC2A40)
Kaique da Silva Ferreira (N671890) (CC2B40)
Matheus Ribeiro de Oliveira (N686EG5) (CC1A40)

Professor(a) Orientador(a):
Myrian

São Paulo
11/2020

Introdução

O seguinte relatório tem como objetivo descrever os processos e etapas da criação da Criptografia de César.

Para a criação da criptografia foi usada a linguagem Python, já que é a linguagem estudada neste semestre em questão.

Desenvolvimento

Objetivos Gerais

O propósito da aplicação de criptografia em uma mensagem, é entender as etapas de criação e funcionamento do código na linguagem, além de adquirir conhecimento do assunto através de pesquisas e estudos.

Metodologia

Para a criação do código utilizamos a linguagem Python, antes de dar início à montagem do script, nos reunimos e planejamos as etapas do programa. Procuramos e pesquisamos, até encontrar algum conteúdo prático e de qualidade, logo após, o grupo foi dividido no período de pesquisas, e recomposto em reuniões semanais, para reunir as informações adquiridas e discutir sobre o trabalho escrito e a formação do código que codifica e descriptografar usando a cifra de César, implementado na linguagem Python.

Utilizamos um IDLE gratuito e online (Repl.it) que permite alteração de até 50 pessoas ao mesmo tempo e com execução do código.

Com o total de 12 reuniões, foi possível compreender as técnicas de Cítala, Atbash, César, Vigenere e Enigma. Que foi suficiente para o desenvolvimento do trabalho teórico e prático, sendo possível até a implementação de funções extras no script do código que está apresentado abaixo:

linhas de código do Programa Cifra de César em Phyton

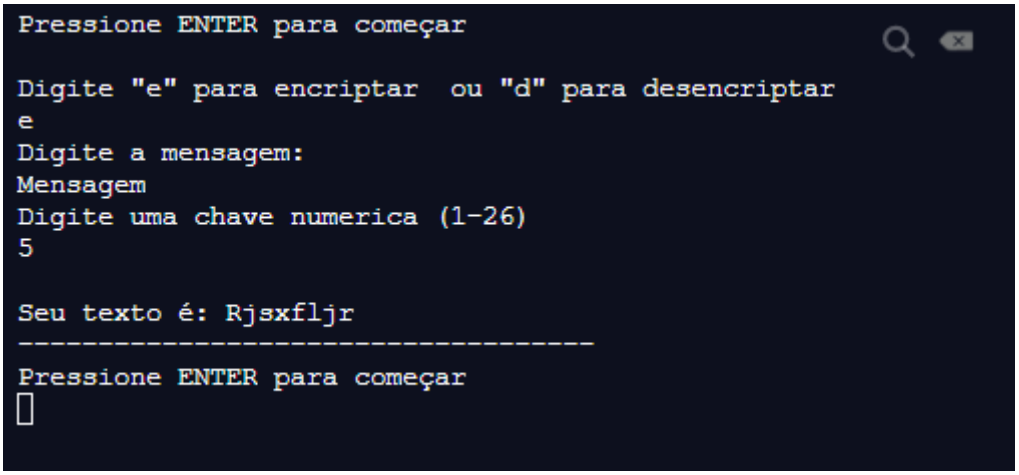
```
#Cifra de César em Phyton

import sys

import os

Tamanho = 26

print('Pressione ENTER para começar')

def reiniciar():

   python = sys.executable

   os.execl(python, python, * sys.argv)

def Modo():

     while True:

        modo = input().lower()

        if modo in 'e d'.split():

            return modo

        else:

            print('Digite "e" para encriptar  ou "d" para desencriptar')

def Mensagem():

    print('Digite a mensagem: ')

    return input()

def Chave():

   chave = 0

   while True:

        print('Digite uma chave numerica (1-%s)' % (Tamanho))

        chave = int(input())

        if (chave >= 1 and chave <= Tamanho):

            return chave
```

```
def Mensagem_Traduzida(modo, mensagem, chave):
    if modo[0] == 'd':
        chave = -chave
    traducao = ''
    for letra in mensagem:
        if letra.isalpha():
            num = ord(letra)
            num += chave
            if letra.isupper():
                if num > ord('Z'):
                    num -= 26
                elif num < ord('A'):
                    num += 26
            elif letra.islower():
                if num > ord('z'):
                    num -= 26
                elif num < ord('a'):
                    num += 26
            traducao += chr(num)
        else:
            traducao += letra
    return traducao
modo = Modo()
Loop = True
while (Loop):
   mensagem = Mensagem()
```

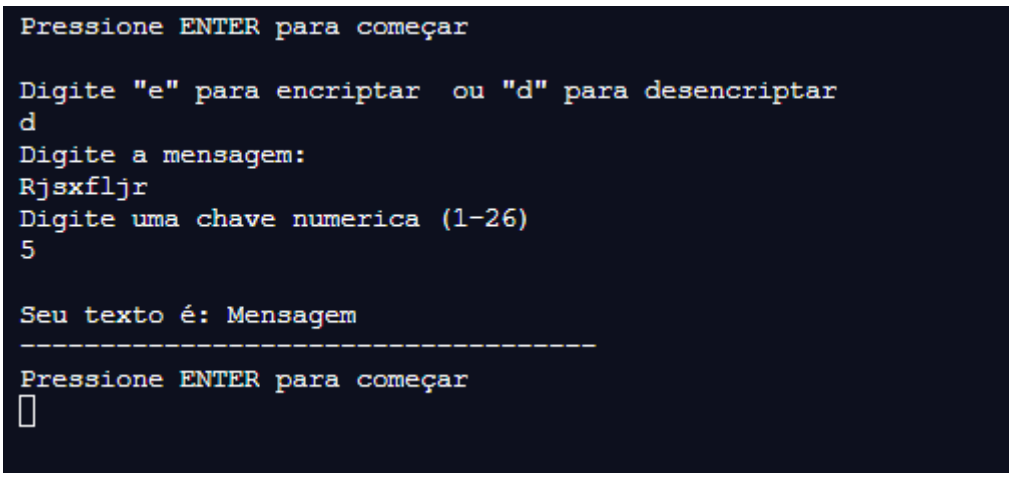
```
    total = 0
    for letras in mensagem:
      total = total + 1
    if total > 128:
      print("-------------------------------------")
      print("Limite de 128 caracteres excedido")
      print("-------------------------------------")
      reniciar= reiniciar()
    else:
      Loop = False
chave = Chave()
print('\nSeu texto é: ', end='')
print(Mensagem_Traduzida(modo, mensagem, chave))
print("-----------------------------------")
reniciar= reiniciar()
```

Programa em Funcionamento

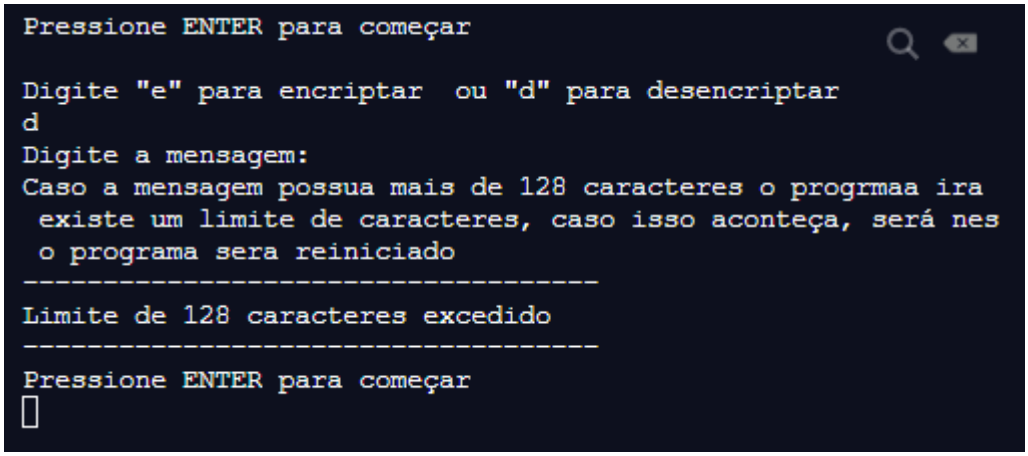
Programa com encriptação de mensagem:



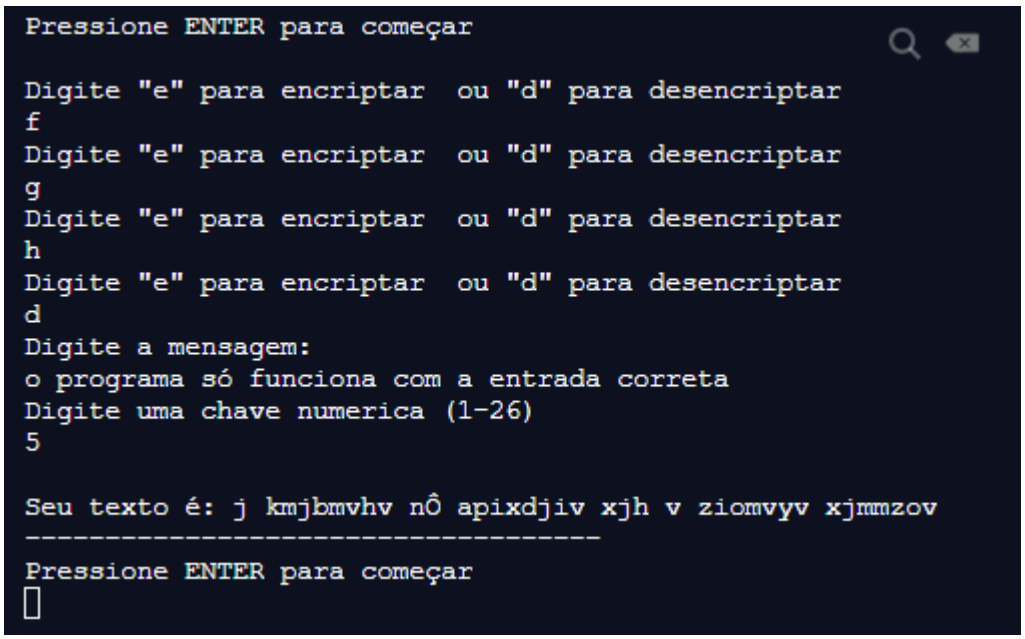
Programa com desencriptação da mensagem:



Programa com números de caracteres excedido:



Quando é digitada a letra errada para encriptar ou desencriptar a mensagem:



Bibliografia

STALLINGS, William. Criptografia e Segurança de Redes. Edição 6. 19 dezembro 2014

http://anaisjem.upf.br/download/op-34-bianchetti.pdf

https://cryptoid.com.br/banco-de-noticias/a-historia-da-criptografia/

https://www.lume.ufrgs.br/bitstream/handle/10183/66106/000870987.pdf

https://blog.validcertificadora.com.br/tipos-de-criptografia-conheca-os-10-mais-usados-e-como-funciona-cada-um/

https://danieldonda.wordpress.com/category/fisica-e-matematica/criptografia/

https://www.docdroid.net/KbpiEqo/criptografia-cifra-de-vigenere-pdf

https://www.cs.du.edu/~snarayan/crypt/vigenere.html

https://danieldonda.wordpress.com/category/fisica-e-matematica/criptografia/

https://youtu.be/6obiuldPcsI

https://www.simonsingh.net/The_Black_Chamber/vigenere_cracking_tool.html

https://www.dcode.fr/vigenere-cipher

https://teses.usp.br/teses/disponiveis/55/55136/tde-06042017-164507/publico/DanieleHelenaBonfim_revisada.pdf

https://www.ime.unicamp.br/~ftorres/ENSINO/MONOGRAFIAS/Fernando_TN17M2.pdf

https://www.lume.ufrgs.br/bitstream/handle/10183/110014/000951896.pdf

https://medium.com/@tarcisioma/algoritmo-de-criptografia-assim%C3%A9trica-rsa-c6254a3c7042

https://cryptoid.com.br/banco-de-noticias/29196criptografia-simetrica-e-assimetrica/

https://www.youtube.com/watch?v=pXu4DV8k8K0

http://campeche.inf.furb.br/tccs/2006-I/2006-1henriquetomasipiresvf.pdf

http://www.ronielton.eti.br/publicacoes/artigorevistasegurancadigital2012.pdf

http://wsmartins.net/ermacs/poster_39.pdf

https://kryptazia.wordpress.com/criptografia/blowfish/

https://www.wikiwand.com/pt/Twofish

http://www.quadibloc.com/crypto/co040301.htm

https://www.youtube.com/watch?v=Tc--MPir6rI

http://clubedosgeeks.com.br/sem-categoria/cifra-de-cesar-criptografia-monoalfabetica

https://www.gta.ufrj.br/grad/10_1/aes/index_files/Page1294.htm

http://wiki.stoa.usp.br/images/c/cf/Stallings-cap2e3.pdf

https://www.youtube.com/watch?v=WuyHJLu4O30

https://teses.usp.br/teses/disponiveis/55/55136/tde-06042017-164507/publico/DanieleHelenaBonfim_revisada.pdf pagina 62

https://www.ime.unicamp.br/~ftorres/ENSINO/MONOGRAFIAS/Fernando_TN17M2.pdf

https://www.lume.ufrgs.br/bitstream/handle/10183/110014/000951896.pdf

https://medium.com/@tarcisioma/algoritmo-de-criptografia-assim%C3%A9trica-rsa-c6254a3c7042

https://cryptoid.com.br/banco-de-noticias/29196criptografia-simetrica-e-assimetrica/

http://campeche.inf.furb.br/tccs/2006-I/2006-1henriquetomasipiresvf.pdf

http://www.ronielton.eti.br/publicacoes/artigorevistasegurancadigital2012.pdf

http://wsmartins.net/ermacs/poster_39.pdf

https://kryptazia.wordpress.com/criptografia/blowfish/

https://www.wikiwand.com/pt/Twofish

http://www.quadibloc.com/crypto/co040301.htm

http://clubedosgeeks.com.br/sem-categoria/cifra-de-cesar-criptografia-monoalfabetica

https://www.gta.ufrj.br/grad/10_1/aes/index_files/Page1294.htm

http://wiki.stoa.usp.br/images/c/cf/Stallings-cap2e3.pdf

http://inventwithpython.com/chapter14.html