Atividade Prática Supervisionada

As técnicas criptográficas seus conceitos, usos e aplicações.

Introdução a programação estruturada.

Lucas G. Francelino

Joao V. Farias

Matheus Ferreira

Sumario

Objetivo do trabalho

Neste trabalho vamos abordar alguns tópicos relacionados a criptografia, que é uma maneira de assegura a privacidade de dados em várias situações dês de uma conversa informal entre amigos, e-mails comerciais, acessos a informações de alunos entre outros.

Vamos apresentar todos os detalhes da criação de um sistema simples de criptografia feito na linguagem pynthon, abordar temas como os fundamentos da criptografia, compara com técnicas atuais, apresentar melhoras etc.

Introdução

A criptografia é uma técnica utilizada há anos que com o passar do tempo evoluiu a ponto de oferecer soluções eficazes no que diz respeito à segurança da informação. Hoje, ela é uma ferramenta de segurança amplamente utilizada nos meios de comunicação e consiste basicamente na transformação de determinado dado ou informação a fim de ocultar seu real significado.

A criptografia pode ser utilizada em aplicações e ambientes cuja segurança das informações é algo relevante para o projeto, principalmente em sistemas WEB, onde o dado trafega em um meio público correndo um risco maior de ser interceptado, fato este que pode gerar prejuízos enormes para uma organização. O domínio das técnicas de criptografia não é algo complexo quando estamos trabalhando com o paradigma orientado a objetos, sendo essencial para a criação de aplicações seguras.

Há pouco tempo, quando a tecnologia ainda não era muito presente em nosso cotidiano, as informações e grande parte dos processos organizacionais eram geridos basicamente no papel, sendo armazenados em armários ou cofres protegidos por cadeados ou senhas.

Vale ressaltar que a criptografia não é aplicada apenas quando um dado é enviado de um local a outro, ela é utilizada também em dispositivos de armazenamento de dados (ex: discos rígidos, pen drives, storages), que são alvos de ataques e roubos. Ou seja, de uma forma geral, a criptografia vai garantir a confidencialidade da informação. Nos próximos tópicos, veremos alguns conceitos relacionados a esta técnica.

Criptografia.

Criptografia em segurança virtual é a conversão de dados de um formato legível em um formato codificado. Os dados criptografados só podem ser lidos ou processados depois de serem descriptografados.

A criptografia é um elemento fundamental da segurança de dados. É a forma mais simples e mais importante de garantir que as informações do sistema de um computador não sejam roubadas e lidas por alguém que deseja usá-las para fins maliciosos.

A criptografia de segurança de dados é amplamente usada por usuários individuais e grandes corporações para proteger as informações dos usuários enviadas entre um navegador e um servidor. Essas informações podem incluir de tudo, desde dados de pagamento até informações pessoais. Os softwares de criptografia de dados, também conhecidos como algoritmo de criptografia ou codificação, são usados para desenvolver um esquema de criptografia que teoricamente pode ser desvendado apenas com uma grande capacidade de processamento.

Quando informações ou dados são compartilhados na Internet, passam por uma série de dispositivos em rede espalhados pelo mundo, que fazem parte da Internet pública. À medida que passam pela Internet pública, os dados correm o risco de serem comprometidos ou roubados por hackers. Para evitar isso, os usuários podem instalar um software ou hardware específico para garantir que os dados ou as informações sejam transferidos com segurança. Esses processos são conhecidos como criptografia em segurança de rede.

A criptografia envolve a conversão de texto simples legível por humanos em texto incompreensível, o que é conhecido como texto cifrado. Essencialmente, isso significa pegar dados legíveis e transformá-los de forma que pareçam aleatórios. A criptografia envolve o uso de uma chave criptográfica, um conjunto de valores matemáticos com os quais tanto o remetente quanto o destinatário concordam. O destinatário usa a chave para descriptografar os dados, transformando-os de volta em texto simples legível. Quanto mais complexa for a chave criptográfica, mais segura será a criptografia, pois é menos provável que terceiros a descritografem por meio de ataques de força bruta (ou seja, tentar números aleatórios até que a combinação correta seja adivinhada).

Existem muitos usos importantes da criptografia

Muitos de nós nos deparamos com a criptografia todos os dias. Os usos populares são:

- Sempre que você usa um caixa eletrônico ou compra algo na Internet por meio de um smartphone, a criptografia é usada para proteger as informações transmitidas.
- Proteção de dispositivos, como criptografia para notebooks.

- Suas mensagens de WhatsApp também são criptografadas, e você também pode ter uma pasta criptografada no seu telefone.
- Seu e-mail também pode ser criptografado com protocolos, como o OpenPGP.
- As VPNs (Redes privadas virtuais) usam criptografia, e tudo que você armazena na nuvem deve ser criptografado. Você pode criptografar todo o seu disco rígido e fazer até chamadas de voz criptografadas.
- A criptografia é usada para provar a integridade e autenticidade das informações usando o que é conhecido como assinaturas digitais. A criptografia é uma parte essencial do gerenciamento de direitos digitais e proteção contra cópias.
- A criptografia pode ser usada para apagar dados. Como as informações excluídas às vezes podem ser recuperadas usando ferramentas de recuperação de dados, se você criptografar os dados primeiro e jogar fora a chave, a única coisa que alguém poderá recuperar será o texto cifrado e não os dados originais.
- **Criptografia simétrica**

O tipo de criptografia simétrica é o mais comum e pressupõe que uma mesma chave usada para ocultar informação precisa ser aplicada para revelá-la na outra ponta. É o tipo de criptografia usada na época da Segunda Guerra Mundial, por exemplo, e protagonista da história da invenção do computador, como conhecemos hoje.

- **Criptografia assimétrica ou de ponta-a-ponta**

Atualmente, os dois protocolos mais usados para proteção de dados na Internet, o SSL (Secure Sockets Layer) e o TLS (Transport Layer Security) utilizam a criptografia simétrica para proteger os dados transmitidos e armazenados.

No entanto, a criptografia simétrica possui um desafio conceitual importante e impossível de ser resolvido. Como combinar uma chave secreta entre duas pessoas que querem se comunicar através da Internet de forma que ela não possa ser obtida por um invasor? Essa pergunta não teve solução até a década de 1970.

A solução foi dada pela criptografia assimétrica, na qual utiliza-se duas chaves distintas, mas que se complementam. Por essa propriedade, dá-se o nome de par de chaves, que é composto pela chave pública e pela chave privada. A chave pública é liberada para todos que desejam se comunicar com o emissor da chave enquanto a chave privada fica em poder de quem a emitiu.

O algoritmo de criptografia mais usados atualmente é o RSA, denominado pelas iniciais dos seus criadores, Ronald Rivest, Adi Shamir e Leonard Adleman. Uma desvantagem dos algoritmos de criptografia assimétrica existentes é o seu desempenho, que são mais lentos que os métodos simétricos.

O mistério da Cicada 3301.

Na Deep Web (internet oculta, que só pode ser acessada por navegadores especiais como o TOR), surgiu em um fórum o que se tornaria um dos maiores mistérios de criptografia da web. Foi postada uma foto com a seguinte mensagem: “Olá. Nós estamos procurando por indivíduos altamente inteligentes. Para encontrá-los, nós criamos um teste. Há uma mensagem oculta dentro desta imagem. Encontre-a, e ela te colocará no caminho para nos encontrar. Estamos ansiosos para conhecer os poucos que conseguirão chegar ao final. Boa sorte. 3301.”. Daí começou uma verdadeira corrida de hackers e pessoas com conhecimentos criptográficos para desvendar tal mistério. Mas a cada descoberta, um novo desafio aparecia – cada vez mais difícil e estranho. Os poucos que continuaram, sumiram misteriosamente dos fóruns.

A cada ano, a Cicada colocava um novo desafio. Muitas teorias foram criadas sobre quem está por trás desses testes, mas a origem é até hoje desconhecida.

Técnicas criptográficas mais utilizadas

A criptografia é uma tecnologia utilizada há bastante tempo para evitar que hackers subtraiam ou fraudem informações sigilosas. Desde a Segunda Guerra Mundial, a máquina de cifração alemã chamada Lorenz era utilizada, mas nos tempos modernos a tecnologia evoluiu e se transformou completamente, originando diferentes tipos de criptografia.

É importante conhecer quais são esses tipos e as suas principais diferenças para entender como tecnologias, por exemplo, certificados digitais, conseguem proteger seus dados. Neste artigo, vamos listar e explicar as 10 criptografias mais usadas e como cada uma delas funciona

Criptografia RC4

A criptografia RC4, sigla para Rivest Cipher 4, é uma cifra de fluxo (stream cipher) criada no fim dos anos 1980, um algoritmo simétrico.Essa cifra opera nos dados um byte por vez, de modo a criptografar esses dados.O RC4 é uma das cifras de fluxo mais usadas, tendo sido usado nos protocolos Secure Socket Layer (SSL) – hoje conhecido como Transport Layer Security (TLS).

Hoje, esse algoritmo não é tão utilizado, pois apresentou algumas vulnerabilidades, que permitiram que usuários quebrassem a chave em questão de um minuto.

Criptografia Twofish

Outro tipo de criptografia simétrica é a Twofish, uma evolução da Blowfish – sendo assim, apenas uma chave de 256 bit é necessária.É bastante útil e segura, sendo finalista de uma competição do Instituto de Tecnologia e Ciências Nacional americano, que buscava uma criptografia para substituir a DES.

Criptografia DES

A criptografia DES, sigla para Data Encryption Standard, é também um tipo de chave simétrica – um dos primeiros que foi criado, datando do começo da década de 1970, por um time de desenvolvedores da IBM.O algoritmo converte texto simples em blocos de 64 bits em texto cifrado, com chaves 48 bits.Por conta do tamanho pequeno da chave, ele é considerado inseguro para várias aplicações atualmente.

Hoje, o DES foi substituído pelo AES.

DESX

Essa é outra variante do DES e trata-se de uma solução bastante simples do algoritmo, mas que aumenta exponencialmente a resistência contra ataques de força bruta sem elevar a sua complexidade computacional.

Basicamente, adicionam-se 64 bits antes da encriptação, o que aumenta a proteção de 120 bits contra força bruta. Atualmente, essa tecnologia não é mais imune contra ataques mais sofisticados, como criptoanálises (o programa evolui a cada tentativa de decifração).

Criptografia 3DES

Derivada do DES, a criptografia 3DES (ou Triplo DES) se tornou popular nos anos 1990 — muito embora hoje já não seja uma unanimidade.

Além disso, vale mencionar, ele se tornará obsoleto a partir de 2023.

Sua diferença para o antecessor é que utiliza 3 chaves de 64 bits.

Criptografia RSA

Já a criptografia RSA é um tipo assimétrico. A sigla diz respeito ao nome de seus criadores, Rivest-Shamir-Adleman.

Ele é muito utilizado hoje em dia e seu funcionamento tem a mesma explicação da criptografia assimétrica.

Ou seja, é baseado na utilização de uma chave pública para criptografar dados e em uma chave privada para descriptografá-los.

Criptografia AES

Já a criptografia AES ou Advanced Encryption Standard é um tipo de cifra que protege a transferência de dados online.É um dos melhores e mais seguros protocolos de criptografia e é utilizado em incontáveis aplicações.Na prática, é uma chave simétrica, pois utiliza a mesma chave para criptografar e descriptografar o conteúdo.Ele também usa o algoritmo SPN (rede de permutação de substituição), aplicando várias rodadas para criptografar dados.

Essas rodadas de criptografia são a razão do alto nível de proteção do AES: se alguém quiser quebrar a criptografia, precisará fazê-lo por várias "rodadas".

Além disso, a criptografia AES conta com 3 tamanhos diferentes de chaves, partindo de 128 bits, 192 bits e 256 bits.

Blowfish

Esse é outro algoritmo desenvolvido para substituir o DES. É uma cifra simétrica que divide as informações em blocos de 64 bits e criptografa cada um deles individualmente.

O *Blowfish* é conhecido por sua velocidade de encriptação e efetividade em geral. Trata-se de uma tecnologia bastante segura, pois há estudiosos no assunto que afirmam que o código não pode ser quebrado.

Ele é completamente grátis, e qualquer indivíduo pode conseguir uma cópia de seu código-fonte, alterar e utilizá-lo em diferentes programas. De forma geral, o *Blowfish* é usado em plataformas de e-commerce para garantir segurança nos pagamentos e proteger senha de acesso dos usuários.

Twofish

O *Twofish* é uma variação do Blowfish e também consiste na cifração de blocos simétricos. A diferença é que ele é formado por blocos de 128 bits e chaves de até 256 bits.

A tecnologia é considerada uma das mais rápidas de seu tipo e é ideal para prover segurança de softwares e hardwares. Seu código-fonte também é gratuito, podendo ser manipulado e utilizado por qualquer programador.

Existe outra variação da mesma criptografia chamada *Threefish*, a diferença é que os tamanhos dos blocos são de 256, 512 e 1024 bits, com chaves do mesmo tamanho.

SAFER

SAFER ("mais seguro" em português) é uma sigla para *Secure and Fast Encryption Routine*. Consiste na criptografia de blocos em 64 bits, por isso é conhecido como *SAFER SK-64*.

Entretanto, foram encontradas fraquezas nesse código, o que resultou no desenvolvimento de novas versões com diferentes tamanhos de chave, como a SK-40, SK-64 e a SK-128 bits.

IDEA

O *Internacional Encryption Algorithm* (IDEA) é uma chave simétrica desenvolvida em 1991, que opera blocos de informações de 64 bits e usa chaves de 128 bits.

O algoritmo utilizado atua de forma diferente, pois usa a confusão e difusão para cifrar o texto. Na prática, ele utiliza três grupos algébricos com operações misturadas, e é dessa forma que o IDEA consegue proteger as informações.

Existem diferentes tipos de criptografia e entendê-los é relevante para que o usuário saiba exatamente como suas informações são protegidas ao utilizar a internet e manusear certificados digitais.

Camellia

Desenvolvido em 2000, *Camellia* é uma criptografia que decifra blocos de informações. Trata-se de uma tecnologia com níveis de segurança bastante semelhantes ao AES, já que pode ser processada em 128, 192 e 256 bits.

Camellia pode ser implementada tanto em softwares (programas) quanto hardwares (peças físicas de computador). Também é compatível com tecnologias mais econômicas de 8 bits (smartcards, sistemas de operação em tempo real etc.) até com processadores mais potentes de 32 bits (computadores de mesa).

CRIPTOGRAFIA QUÂNTICA

Este tipo de codificação de informação é diferente dos demais métodos criptográficos porque não precisa do segredo nem da conta to prévio entre as partes. A criptografia quântica permite a detecção de intrusos e é totalmente segura mesmo que o intruso tenha poder computacional ilimitado. Mas o seu custo de implantação é muito alto. Outro fato limitante para o uso dessa técnica é a taxa de erros na transmissão dos fótons, seja por ondas de rádio ou fibra ótica. Até agora, os melhores resultados foram obtidos por meio de fibras de altíssima pureza, abrangendo uma distância de cerca de 70 km.

REDES SEM FIO

As senhas da rede sem fio são criptografadas de forma a permitir a navegação somente para quem informar a senha correta. Porém, abriram uma grande possibilidades de interceptação de dados e roubo de conexões. As tecnicas mais, usadas na criptografia de rede wireless são WEP, WPA e WPA2.

WPA e WPA2

Surgiu em 2003 de um esforço conjunto de membros da Wi-fi Aliança e de membros de IEEE(instituto de Engenheiros Eletricistas e Eletronicos), empenhados em aumentar o nível de segurança das redes wireless. A WPA fornece criptografia para empresas, e a WPA2-considerada a próxima geração de segurança sem fio vem sem usada por muitos orgãos governamentais em todo o mundo.

Estruturação, conceitos e fundamentos

A técnica de criptografia escolida foi a de substituição de caracteres por números com, trocando estes números já estabelecidos na tabela asci, a criptografia acontece devido ao embaralhamento de caracteres com uma conta matemática e depois é somado, é retornado ao usuário a frase já descriptografada.

```
**************************************************************************
  Para criptografar digite 'c' e para descriptografa digite 'd'.
**************************************************************************
O que deseja fazer? c
Escreva a mensagem para criptografa: ola mundo
**************
  c`U aibXc
--------------
```

Assim como a Cifra de César, que igualmente é uma criptografia matemática, o código abaixo é um exemplo de uma conta matemática que faz a operação perante uma biblioteca já existente usando posições de caracteres por números.

```
def crip():
    palavra = input("Escreva a mensagem para criptografa: ")
    msg = ''
    for i in palavra:
        msg = msg + chr(ord(i) + 10 - 2 + 15 - 35) # O 'ord' ele transforma letra em numero pela tabela asci.
    aparencia(msg)  # Já o 'chr' tranforma decimal em letra.
```

Técnica de descriptografia

A técnica de descriptografia é basicamente a mesma técnica de criptografia, o que muda na fórmula é que invés de fazer uma conta de adição à biblioteca, será uma conta de subtração com os mesmos elementos de criptografia, a chave de criptografia será uma conta inversa à biblioteca, fazendo assim mais segura pois, a segurança é definida à quem tem a biblioteca de caracteres correta.

```
def des():
        palavra = input("Escreva a mensagem para descriptografa: ")
        msg = ''
        for i in palavra:
            msg = msg + chr(ord(i) - 10 + 2 - 15 + 35)
        aparencia(msg)
```

```
**********************************************************************
  Para criptografar digite 'c' e para descriptografa digite 'd'.
**********************************************************************
O que deseja fazer? d
Escreva a mensagem para descriptografa: c`U□aibXc
**************
  ola mundo
**************
```

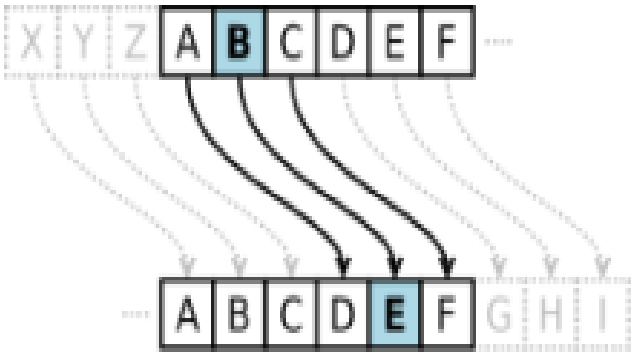
O método utilizado teve o conceito em criptografias rápidas e simples, mas seguras, como na

Roma antiga, Cifra de César, também conhecida como cifra de troca, código de César ou troca de César, é uma das mais simples e conhecidas técnicas de criptografia. É um tipo de cifra de substituição na qual cada letra do texto é substituída por outra, que se apresenta no alfabeto abaixo dela um número fixo de vezes. Por exemplo, com uma troca de três posições, A seria substituído por D, B se tornaria E, e assim por diante. O nome do método é em homenagem a Júlio César, que o usou para se comunicar com os seus generais.

O processo de criptografia de uma cifra de César é frequentemente incorporado como parte de esquemas mais complexos, como a cifra de Vigenère, e continua tendo aplicações modernas, como no sistema ROT13.

Segundo Tanenbaum e Wetherall (2011, p. 4 83) proteger suas correspondências particulares e de cunho militar um código de substituição no qual cada letra do texto original era substituída pela letra que está três posições a frente dela no a alfabeto: a letra "a" era substituída por "d", a "b" "e", e assim sucessivamente, criptografia usa um sistema simples de substituição monoalfabética, onde é substituída por outra letra de acordo com a chave utilizada, podendo ser ela qualquer uma das 26 letras do alfabeto, criando assim um texto criptografado, mantendo a ordem dos símbolos do texto original.

A cifra de César usa o que chamamos de criptografia de chaves simétricas, onde a mesma chave usada para criptografar o texto é usada também para descriptografar.



Benefícios em relação as técnicas anteriores

As técnicas utilizadas, por meio das funções, permitem que o código não se repita e que se torne composto por pequenas rotinas pré determinadas.

As funções, trazem ao programa procedimentos que podem ser utilizados em varias partes do código de forma equivalente, sempre mantendo a estrutura original.

Sendo assim, trazemos ao código com essa técnica versatilidade e assiduidade, para que possa ser realizadas manutenções em apenas um lugar do código e execuções de diferentes tarefas em lugares distintos sem precisar repetir novamente a tarefa necessária.

```
def criptografia():
    mensagem = input("Digite a Mensagem para criptografar:\n")
    cripto = ""

    for letra in mensagem:
        indice = alfabeto.index(letra)
        substituicaoletra = alfabeto[(indice + valormud) % n]

        cripto = cripto + substituicaoletra



    print('A sua mensagem critpografada é:',cripto)
    continuacao()
```

Dessa maneira, dentro das funções foram utilizadas outras técnicas como:

Arrays, em que foi criado a própria tabela de caracteres, trazendo como beneficio ao código a singularidade, dado que apenas as pessoas com conhecimento de como está ordenada os valores da lista conseguirá decifrar ou quebrar a criptografia. Podemos verificar a lista na fig 5.2.2:

```
alfabeto = ['a','A','@', 'b','B','Õ', '$', 'c','C', '#', 'd','D','á','í','$','Í',
            'e','E', '%', 'f','F', 'Ī', 'g','G', '&', 'h','H', '*', 'i',
            'I','(', 'j','J', ')', 'k','K','-','é','ê','ë','Ù','Á','É',
            'l','L', '/', ':','ã', 'm','M','ì', '[', 'n','N', ']', 'o','O', '=',
            'p','P','â', '+', 'q','Ò','Q','ì', 'Ü', 'r','R','Ê','^', 's','S','ú','Ó',
            't','T', '"', 'u','U', 'ã', 'v','Ø','Õ','V', 'w','W','x','X','ó', 'y',
            'Y', 'z','Z','Ī','Ü', '0', '|', '1','Ã', '~', '2', '3','ô', '4', '>','5','Ü',
            'ó', '<', '7', '8','ô', '9', '_', ',', '-', '?', '!', ' ','Ã']

valormud = 9
```

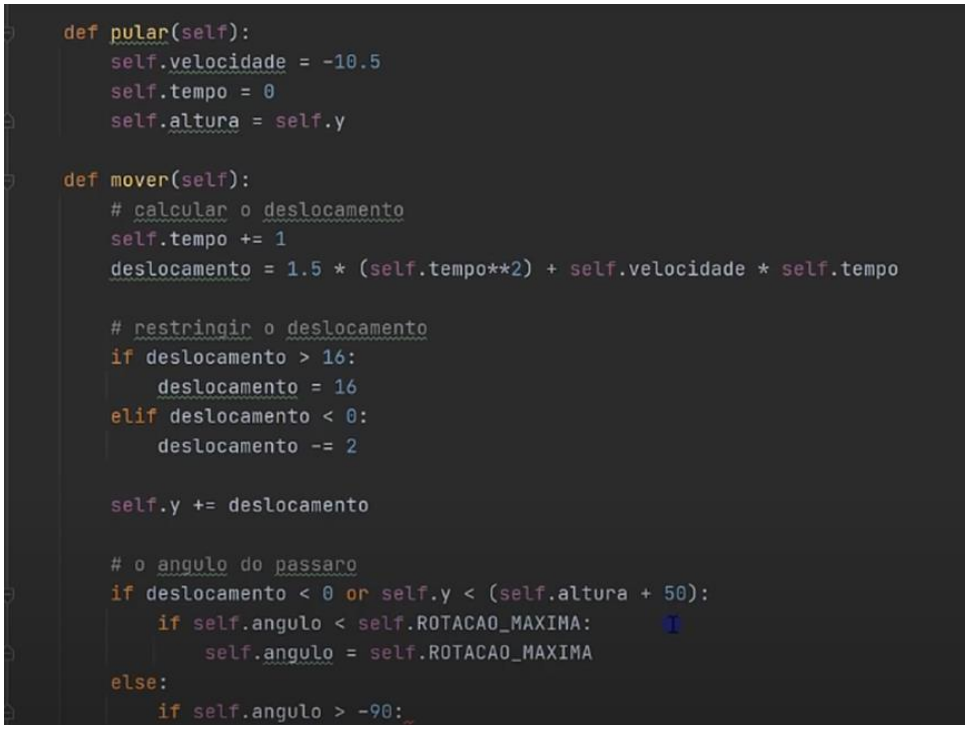
Percebemos, que a lista não segue nenhum tipo de padrão pré-definido anteriormente, e que também utilizamos uma constante para definir o valor que iremos alterar a informação recebida, isto é, carácter por carácter irá nove casas a frente do seu valor original na lista.

Dessa forma, utilizamos o loop da técnica “for in” para verificar carácter por carácter e fazer a alteração com a constante, o que trás agilidade para o código. Não somente, mas também utilizamos a técnica de loop para que o código continue sendo executado e que se mantenha ativo de acordo com as necessidades do usuário. Consequentemente, todas essas técnicas aplicadas em conjuntas fazem o programa se tornar ágil, prático de ser executado e que abrange a necessidade do usuário de utilizar o programa quantas vezes achar necessários.

Aplicações que fazem ou fizeram uso da tecnica

Atualmente, são varias as aplicações que utilizam funções como técnica para construir seu código, graças aos grandes benefícios que essa metodologia trás. No inicio de 2014 um pequeno jogo para smartphones chamado Flappy Bird fez grande sucesso.
Apesar do grande sucesso de Flappy Bird, o mesmo foi programado utilizando funções básicas e que foram definidas para serem executadas durante toda a execução do jogo.

Na fig 5.3.1, mostra como essa técnica em python pode ser empregada dentro do jogo Flappy Bird:



```
def pular(self):
    self.velocidade = -10.5
    self.tempo = 0
    self.altura = self.y

def mover(self):
    # calcular o deslocamento
    self.tempo += 1
    deslocamento = 1.5 * (self.tempo**2) + self.velocidade * self.tempo

    # restringir o deslocamento
    if deslocamento > 16:
        deslocamento = 16
    elif deslocamento < 0:
        deslocamento -= 2

    self.y += deslocamento

    # o angulo do passaro
    if deslocamento < 0 or self.y < (self.altura + 50):
        if self.angulo < self.ROTACAO_MAXIMA:
            self.angulo = self.ROTACAO_MAXIMA
    else:
        if self.angulo > -90:
```

Dessa forma, podemos perceber com a imagem que o jogo “Flappy Bird” está utilizando a definição de funções para deixar pré definidas tarefas a serem executadas no código.

Não somente, mas também é utilizado dentro da aplicação “Flappy Bird” a técnica “For In”, trazendo então um loop para o programa. Este loop é aplicado no programa para trazer uma variedade de seleção de diferentes “pássaros” para o usuário.

Na fig 5.3.2, mostra como é empregado o “For In” no código do jogo.

```
def main():
    passaros = [Passaro(230, 350)]
    chao = Chao(730)
    canos = [Cano(700)]
    tela = pygame.display.set_mode((TELA_LARGURA, TELA_ALTURA))
    pontos = 0
    relogio = pygame.time.Clock()

    rodando = True
    while rodando:
        relogio.tick(30)

        for evento in pygame.event.get():
            if evento.type == pygame.QUIT:
                rodando = False
                pygame.quit()
                quit()
            if evento.type == pygame.KEYDOWN:
                if evento.key == pygame.K_SPACE:
                    for passaro in passaros:
                        passaro.pular()
```

E por fim, podemos notar na FIG 5.3.2: Também foi utilizado a definição de lista para armazenar diferentes tipos de dados.

Discursão comparativas entre estas técnicas e outras conhecidas

SIMETRICA

Esse tipo de criptografia também é chamado de criptografia de chave única, criptografia de chave privada, criptografia de chave compartilhada, criptografia de chave secreta ou criptografia de chave convencional. Vale ressaltar que nem sempre temos apenas uma chave aqui, porém a outra chave é fortemente baseada na primeira. Dessa forma, podemos ter uma chave para cifragem e uma chave para decifragem que seja diferente da primeira, mas baseada naquela. Como um

exemplo prático podemos imaginar que a primeira chave poderia ser "Roma" e a segundo "amor", ou seja, o contrário da primeira, mas facilmente deduzível.

Normalmente também temos que a chave de cifragem é igual a chave de decifragem.

A transformação é dada caractere por caractere ou bit a bit. A execução é mais rápida se comparada a criptografia de chave assimétrica.

Utilizando a criptografia simétrica temos como garantia a confidencialidade e a integridade. A irretratabilidade e a autenticidade não são garantidas.

Basicamente temos como principais problemas na criptografia simétrica manter o sigilo da chave e a dificuldade no compartilhamento da chave devido os problemas de segurança nos canais de comunicação. Segue na Figura 1 um esquema de como funciona basicamente a criptografia simétrica.

A chave geralmente é um número pequeno de até 256 bits e utilizamos o RNG (Random Number Generator) ou PRNG (Pseudo Random Number Generator) para a sua geração. Utilizando o RNG temos a geração de números aleatórios em que não há como repetir o processo ou podemos utilizar o PRNG onde temos pseudos números aleatórios gerados em que conseguimos gerar o mesmo número se tivermos a mesma situação que gerou o número anterior. Neste caso o processo de gerenciamento de chaves é complexo. Em média para se quebrar uma chave de 40 bits ao custo de 100 mil dólares têm-se um tempo estimado no total de dois segundos para quebrá-la, porém se tivermos uma chave de 128 bits e gastarmos 10 trilhões de dólares em recursos para quebrar essa chave teremos um total estimado de dez elevado na décima primeira potência (10^{11}) de anos para quebrá-la. Esses são estudos comprovados baseando-se em tecnologia atual e de ponta.

Entre os sistemas criptográficos simétricos temos: IDEA, TwoFish, BlowFish, Serpent, DES, AES, RC5, RC6. Esses também são chamados de criptografia "De Bloco" ou "Baseada em Bloco". Nesse tipo de criptografia são processados blocos de informação de uma só vez, concatenando-os no final do processo. Outros dois sistemas criptográficos são o RC4 e OTP que são chamados criptografia "De Fluxo" ou "Baseada em Fluxo" em que nesse tipo de criptografia é processado cada bit da mensagem individualmente (processamento bit a bit).

Nas próximas seções veremos um pouco mais sobre as criptografias de bloco destacando o AES e o DES e na sequencia veremos a criptografia de fluxo destacando o RC4 e OTP.

A criptografia simétrica é a mais antiga e mais conhecida das técnicas de encriptação. Os dados são divididos em blocos, e uma chave secreta é aplicada a cada um deles, alterando a ordem das letras ou as substituindo por números para criar o texto cifrado. Essa técnica usa somente uma chave secreta para cifrar e decifrar os dados, de maneira que, se o destinatário não a tiver, será necessário enviá-la separadamente.

Prós e contras

Se ambos o remetente e o destinatário tiverem a chave secreta, eles poderão cifrar e decifrar todas as mensagens que a usam, o que é ao mesmo tempo uma vantagem e uma vulnerabilidade do sistema. A criptografia simétrica é de implementação rápida e fácil, o que a faz a torna a forma de encriptação mais comum em transações de compra e venda online. Porém, se a chave for interceptada por um invasor, ele terá o que precisa para decifrar todas as mensagens que usam essa chave. Algoritmos da criptografia simétrica também tendem a ser mais simples -- e, portanto, mais fáceis de entender e decodificar -- do que os algoritmos da criptografia assimétrica.

Criptografia assimétrica

A criptografia assimétrica, também conhecida como criptografia de chave pública, usa duas chaves relacionadas entre si: uma pública, para cifrar os dados, e uma privada, para decifrá-los. Informações em forma de texto são tratadas como números imensos, que são elevados a potência de um segundo número imenso e, então, divididos por um terceiro número, gerando um produto final que será novamente convertido em texto, desta vez criptografado.

Vunerabilidades e Falhas

Prós e contras

Os algoritmos da criptografia assimétrica são mais complexos do que os da simétrica, portanto são mais lentos e necessitam de mais poder de processamento. No entanto, por isso também são muito mais seguros. A chave pública pode ser distribuída a qualquer um que possa ter interesse em criptografar uma mensagem, mas a chave privada nunca é divulgada, o que não a deixa suscetível a invasores. Os dados só podem ser cifrados com a chave pública e

decifrados com a chave privada, o que significa que uma vez feita a criptografia, nem o remetente pode decifrá-la sem uma chave privada.

É importante conhecer quais são esses tipos e as suas principais diferenças para entender como tecnologias, por exemplo, certificados digitais, conseguem proteger seus dados. Vamos listar e explicar as criptografias mais usadas e como cada uma delas funciona.


security key

Melhorias propostas ou implementadas

Atualmente no mundo que vivemos a tecnologia cada vez está tomando tudo em nossa volta, cada pequeno gesto que você faz no seu dia a dia pode estar ligado diretamente ou indiretamente a algo tecnológico, com isso a internet um fruto da tecnologia que também vem cada vez crescendo, hoje pode se ver que quase todos os meios de comunicação. Como quais tudo está ligado a internet necessitamos de algo seguro para que podemos usufruir dela da melhor forma, com isso atualmente quase tudo que usamos tem criptografia de ponta a ponta para que nossa experiencia com a internet seja boa e segura experiência desde jogar algo pela net, enviar um e-mail para seu chefe ou postar aquela foto no feed do Instagram, tudo disso e necessário ter criptografia, ela é uma grande melhoria no ramo de tecnologia para que pessoas mal intencionados os famosos (hackers) não possam rouba alguma informação sigilosa para seu beneficio ou de terceiros. A criptografia é um das melhores formas de prevenir invasão de segurança de algo na net, a criptografia está presente e quais tudo até no seu celular, por exemplo no whatzapp, para que por exemplo você está conversando com sua namorada, para que só você e ela receba as mensagens que estão digitado e que ninguém mais possa ver é usado uma criptografia de ponta a ponta fazendo com que seja quase impossível de se hacker as informações digitadas na conversa entre você e sua namorada, mas não só no whatzapp que tem criptografia como já dito em quase tudo hoje e necessário ter criptografia.

Projeto (Estrutura do programa)

- Em modo geral essa e estrutura do trabalho:

```
def crip():
    palavra = input("Escreva a mensagem para criptografa: ")
    msg = ''
    for i in palavra:
        msg = msg + chr(ord(i) + 10 - 2 + 15 - 35)
    print(msg)
def des():
        palavra = input("Escreva a mensagem para descriptografa: ")
        msg = ''
        for i in palavra:
            msg = msg + chr(ord(i) - 10 + 2 - 15 + 35)
        print(msg)
while True:
    print("Para criptografar digite 'c' e para descriptografa digite 'd'. ")
    opcao = str(input('O que deseja fazer? '))
    if opcao == 'c':
        crip()
    elif opcao == 'd':
        des()
    else:
        print("Digite uma opção válida.\n")
```

```
print("Para criptografar digite 'c' e para descriptografa digite 'd'. ")
opcao = str(input('O que deseja fazer? '))
if opcao == 'c':
    crip()
elif opcao == 'd':
    des()
else:
    print("Digite uma opção válida.\n")
continua = str(input("Deseja criptografar ou descriptografa outra palavra? digite 's' ou 'n': "))
if continua == 's':
    print('OK VAMOS CINTINUA!! ')
elif continua == 'n':
    print(' FIM!! ')
    break
else:
    print("Digite uma opção válida.\n")
```

Explicação cada parte:

Parte 1

- Essa parte e a última escrita na linha de código mais e a primeira a ser executada como ela e chamada as outras partes do programa.
- Ela começa com um loop infinito usando o While True.
- Também nessa parte usamos condições If, elif e else e também usamos o break para poder parar o código.

```python
# Inicialização do codigo.
while True:
    print("Para criptografar digite 'c' e para descriptografa digite 'd'. ")
    opcao = str(input('O que deseja fazer? '))
    if opcao == 'c':
        crip()
    elif opcao == 'd':
        des()
    else:
        print("Digite uma opção válida.\n")
    continua = str(input("Deseja criptografar ou descriptografa outra palavra? digite 's' ou 'n': "))
    if continua == 's':
        print('OK VAMOS CINTINUA!! ')
    elif continua == 'n':
        print(' FIM!! ')
        break
    else:
        print("Digite uma opção válida.\n")
```

Parte 2

- Essa e primeira parte escrita na linha de código mais e a segunda usada, quando o programa e usado.
- Nessa parte ela é uma função como na print pode se ver ela e responsável pela criptografia do que for desejado criptografa.
- Nessa parte contém um loop usando for.
- Também contém chr e ord que na print também explica sua funcionalidade.

```python
# Parte do codigo responsavel pela criptografia
def crip():
    palavra = input("Escreva a mensagem para criptografa: ")
    msg = ''
    for i in palavra:
        msg = msg + chr(ord(i) + 10 - 2 + 15 - 35)  # O 'ord' ele transforma letra em numero pela tabela asci.
    print(msg)  # Já o 'chr' tranforma decimal em letra.
```

Parte 3

- Já essa parte e a segunda escrita na linha de código mais e última acionada no programa, sé caso o usuário deseja descriptografa algo.
- Essa parte também é uma função.
- Nela e usada um loop for também.

- E mais uma vez e usado o chr e ord que explicado na print anterior.

```python
# Parte do codigo responsavel pela descriptografia
def des():
        palavra = input("Escreva a mensagem para descriptografa: ")
        msg = ''
        for i in palavra:
            msg = msg + chr(ord(i) - 10 + 2 - 15 + 35)
        print(msg)
```

Relatório com as linhas de codigo

```
def crip():
    palavra = input("Escreva a mensagem para criptografa: ")
    msg = ''
    for i in palavra:
        msg = msg + chr(ord(i) + 10 - 2 + 15 - 35)
    print(msg)
def des():
        palavra = input("Escreva a mensagem para descriptografa: ")
        msg = ''
        for i in palavra:
            msg = msg + chr(ord(i) - 10 + 2 - 15 + 35)
        print(msg)
while True:
    print("Para criptografar digite 'c' e para descriptografa digite 'd'. ")
    opcao = str(input('O que deseja fazer? '))
    if opcao == 'c':
        crip()
    elif opcao == 'd':
        des()
    else:
        print("Digite uma opção válida.\n")
```

```
print("Para criptografar digite 'c' e para descriptografa digite 'd'. ")
opcao = str(input('O que deseja fazer? '))
if opcao == 'c':
    crip()
elif opcao == 'd':
    des()
else:
    print("Digite uma opção válida.\n")
continua = str(input("Deseja criptografar ou descriptografa outra palavra? digite 's' ou 'n': "))
if continua == 's':
    print('OK VAMOS CINTINUA!! ')
elif continua == 'n':
    print(' FIM!! ')
    break
else:
    print("Digite uma opção válida.\n")
```

```python
# Inicialização do codigo.
while True:
    print("Para criptografar digite 'c' e para descriptografa digite 'd'. ")
    opcao = str(input('O que deseja fazer? '))
    if opcao == 'c':
        crip()
    elif opcao == 'd':
        des()
    else:
        print("Digite uma opção válida.\n")
    continua = str(input("Deseja criptografar ou descriptografa outra palavra? digite 's' ou 'n': "))
    if continua == 's':
        print('OK VAMOS CINTINUA!! ')
    elif continua == 'n':
        print(' FIM!! ')
        break
    else:
        print("Digite uma opção válida.\n")
```

```python
# Parte do codigo responsavel pela criptografia
def crip():
    palavra = input("Escreva a mensagem para criptografa: ")
    msg = ''
    for i in palavra:
        msg = msg + chr(ord(i) + 10 - 2 + 15 - 35)  # O 'ord' ele transforma letra em numero pela tabela asci.
    print(msg)  # Já o 'chr' tranforma decimal em letra.
```

```python
# Parte do codigo responsavel pela descriptografia
def des():
        palavra = input("Escreva a mensagem para descriptografa: ")
        msg = ''
        for i in palavra:
            msg = msg + chr(ord(i) - 10 + 2 - 15 + 35)
        print(msg)
```

Bibliografia

Fonte técnicas mais conhecidas: https://cryptoid.com.br/