Tutorial: Criptografia de Arquivo de Texto com Python

Neste tutorial, vamos aprender como utilizar o Python para cifrar e decifrar arquivos de texto utilizando o módulo cryptography. Vamos usar o algoritmo AES (Advanced Encryption Standard) para isso.

Objetivo:
- Utilizar o módulo cryptography do Python para cifrar e decifrar arquivos de texto.
- Compreender os conceitos básicos de criptografia simétrica e a importância da chave de criptografia.

Passo 1: Instalação do módulo cryptography

Crie um ambiente virtual python para este exercício.

Se você ainda não tiver o módulo cryptography instalado, pode instalá-lo utilizando o pip:

```
pip install cryptography
```

Passo 2: Importação do Módulo e Geração da Chave de Criptografia

Vamos começar importando o módulo cryptography e gerando uma chave de criptografia. A chave é essencial para cifrar e decifrar o arquivo.

```
from cryptography.fernet import Fernet

# Geração da chave de criptografia
chave = Fernet.generate_key()
cipher_suite = Fernet(chave)
```

Passo 3: Cifragem do Arquivo de Texto

Agora, vamos cifrar um arquivo de texto chamado `arquivo.txt`. Crie um arquivo com esse nome e com um conteúdo escolhido por você na mesma pasta do seu script python.

Primeiro, abrimos o arquivo para leitura, lemos seu conteúdo e depois ciframos esse conteúdo.

```
with open('arquivo.txt', 'rb') as arquivo:
    texto = arquivo.read()
    texto_cifrado = cipher_suite.encrypt(texto)

# Salva o texto cifrado em um novo arquivo
with open('arquivo_cifrado.txt', 'wb') as arquivo_cifrado:
    arquivo_cifrado.write(texto_cifrado)
```

Passo 4: Decifragem do Arquivo Cifrado

Para decifrar o arquivo cifrado e recuperar o texto original, utilizamos a mesma chave de criptografia.

```
with open('arquivo_cifrado.txt', 'rb') as arquivo_cifrado:
    texto_cifrado = arquivo_cifrado.read()
    texto_decifrado = cipher_suite.decrypt(texto_cifrado)

# Salva o texto decifrado em um novo arquivo
with open('arquivo_decifrado.txt', 'wb') as arquivo_decifrado:
    arquivo_decifrado.write(texto_decifrado)
```

Passo 5: Verificação do Arquivo Decifrado

Verifique se o arquivo decifrado `arquivo_decifrado.txt` contém o texto original.

Desafios para reflexão

Observe que o código usa a mesma chave para criptografar e descriptografar o arquivo. Isso não é muito realista. Geralmente temos um programa que criptografa a informação e outro programa que decifra. Pensando nisso, realize os exercícios a seguir.

1. Crie dois scripts separados python e pense em alguma forma de compartilhar a chave secreta entre os dois programas. Tente usar pelo menos uma das estratégias a seguir:
    a. abrir um prompt no terminal e digitar/colar a chave
    b. transferir usando sockets
    c. fazer com que os dois códigos leiam a chave a partir de um arquivo de texto

2. Agora use uma lógica parecida não para transferir a chave, mas para transferir o arquivo. Um código deve criptografar o arquivo e abrir um socket. O outro código deve ler o arquivo pelo socket e decifrar o arquivo.

3. Use algoritmos com tamanhos de chave diferentes. Você deve informar o algoritmo e os parâmetros na variável cipher_suite. Pesquise como fazer isso na documentação da biblioteca Cipher. O tempo de processamento aumenta ou diminui?

4. Pesquise como alterar o tamanho do bloco de criptografia do algoritmo. Pesquise como fazer isso na documentação da biblioteca Cipher.Realize experimentos alterando o tamanho do bloco e observe: o tempo de criptografia aumentou ou diminuiu?

5. Refaça os exercícios anteriores com arquivos grandes, de muitos MB ou até mesmo GB. Observe o tempo de criptografia.

Conclusão

Neste tutorial, você aprendeu como utilizar os módulos cryptography e Cipher do Python para cifrar e decifrar arquivos de texto. A criptografia simétrica é uma técnica importante para proteger informações sensíveis, garantindo que apenas pessoas autorizadas possam acessá-las. Experimente cifrar e decifrar diferentes arquivos e explore outras opções e parâmetros do módulo cryptography para expandir seu conhecimento sobre criptografia.