



دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)

گزارش پروژه اول درس امنیت اطلاعات

دانشجو: مرتضی صفری ۹۸۳۱۰۳۹

استاد: دکتر شهریاری

آبان ۱۴۰۲

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

فاز اول

این فاز شامل دو بخش است.

بخش اول

در این بخش با توجه به استانداردها رمز عبوری وارد شده توسط کاربر مورد بررسی قرار میگیرد و در صورت ناامن بودن آن علت ناامن بودن آن برای کاربر چاپ می شود. کد آن به صورت زیر است:

```
import re

def check_password_strength(password):
    if len(password) < 8:
        return "Password should be have at least 8 character..."

    with open('dictionary.txt', 'r') as f:
        dictionary = f.read().splitlines()

    if password in dictionary:
        return "This password is common and cannot be accepted..."

    if not re.search(r'[A-Z]', password) or not re.search(r'[a-z]', password):
        return "Password should contain uppercase and lowercase letters..."

    if not re.search(r'\d', password):
        return "Password should contain 1 or more digit..."

    if not re.search(r'[@#$%^&*(),.?":{}|<>_-]', password):
        return "Password must contain symbols..."

    return "Password is strength."

password = input("Please enter your password: ")
result = check_password_strength(password)
print(result)
```

در این کد ۵ شرط بررسی می‌شود. شرط اول بررسی می‌کند که طول رمز کمتر از ۸ کاراکتر نباشد. شرط دوم به بررسی وجود رمز ورودی کاربر در دیکشنری که خودمان آن را ایجاد کرده ایم می‌پردازد. دیکشنری در واقع بیانگر این است که چه رمز های رایجی مورد استفاده قرار می‌گیرد و درون شرط ما مانع انجام این کار می‌شود تا از حمله دیکشنری جلوگیری شود. شرط سوم به لزوم وجود کلمات توأم با حروف بزرگ و کوچک می‌پردازد. شرط چهارم بررسی وجود عدد در رمز عبور را بررسی می‌کند و شرط پنجم وجود کاراکتر را در ورودی کاربر بررسی می‌کند.

در نهایت اگر رمز عبور موردی نداشت عبارت "رمز عبور قوی است" چاپ می‌شود.

بخش دوم

در این بخش به حدس یک رمز عبور با توجه به برخی اطلاعات می‌پردازد. کد بنده در دو مود پیاده سازی شده است. مود استاندارد که فقط با استفاده از تعداد کاراکتر ها تصمیم می‌گیرد و مود دوم که حرف اول را میداند. کد آن به صورت زیر است:

```
import math
import time
global guess

pasw = str(input('Input password: '))
chars = 'abcdefghijklmnopqrstuvwxyz' #only limited myself to lowercase for
simplicity.
base = len(chars) + 1
size = int(input('enter size of password: '))
mod = int(input('enter mod number (1-standard, 2-with first character, 3-with
part of pass): '))
firstChar = ''
if mod==2:
    firstChar = str(input('enter first character: '))
    size-=1
t1 = time.time()
```

```

#mod = input('please enter number of mod: ')
def cracker(pasw):
    guess = ''
    temp=0
    for i in range(size-1):
        temp += pow(base, i)
    print(temp)
    tests = pow(base,size-1) + temp
    c = 0
    m = 0

    while True:
        y = tests
        while True:
            c = y % base
            m = math.floor((y - c) / base)
            y = m
            guess = chars[(c - 1)] + guess
            if m == 0:
                break
        guess = firstChar + guess
        print(guess)
        if guess == pasw:
            t2 = time.time()
            print('Got "{}" after {} tests'.format(guess, str(tests-
pow(base,size-1) - temp + 1)))
            print('Time of crack password was {}'.format(t2 - t1))
            break
        else:
            tests += 1
            guess = ''

cracker(pasw)
input()

```

به بررسی جز به جز آن میپردازیم.

```
base = len(chars) + 1
```

در اینجا برای بررسی index حروف الفبا و حدس حروف base به علاوه یک شده است.

```
t1 = time.time()
```

در اینجا تایم ابتدایی در متغیری ذخیره میشود.

```

def cracker(pasw):
    guess = ''
    temp=0
    for i in range(size-1):
        temp += pow(base, i)
    print(temp)
    tests = pow(base,size-1) + temp
    c = 0
    m = 0

    while True:
        y = tests
        while True:
            c = y % base
            m = math.floor((y - c) / base)
            y = m
            guess = chars[(c - 1)] + guess
            if m == 0:
                break
        guess = firstChar + guess
        print(guess)
        if guess == pasw:
            t2 = time.time()
            print('Got "{}" after {} tests'.format(guess, str(tests-
pow(base,size-1) - temp + 1)))
            print('Time of crack password was {}'.format(t2 - t1))
            break
        else:
            tests += 1
            guess = ''

```

این بخش اصلی کد است و با توجه به آن مقدار پسورد حدس زده می‌شود.

```

for i in range(size-1):
    temp += pow(base, i)

```

این تیکه کد برای این است که دقیقاً جایی کار حدس زدن را شروع کنیم تا از اولین عبارات n حرفی که سائز آن توسط کاربر وارد شده است، باشد. به عنوان مثال ما برای شروع بررسی اولین عبارت دو حرفی باید از `index`

۲۷ شروع کنیم چون تعداد حروف الفبا ۲۶ تا است. و برای اولین کلمه سه حرفی از $۲۶ * ۲۶ + ۱$ شروع میکنیم.

```
guess = firstChar + guess
```

این قطعه کد برای این است که اگر در مود ۲ باشیم، حرف اول را اضافه کند. توجه شود که قبل از آن باید از مقدار ساینز یکی کم کرد.

```
print('Got "{}" after {} tests'.format(guess, str(tests-  
pow(base,size-1) - temp + 1)))  
print('Time of crack password was {}'.format(t2 - t1))
```

این قطعه برای نمایش تعداد حدس ها و زمان سپری شده تا رسیدن به مقدار درست را چاپ میکند. برای تعداد توجه شود آن مقداری که ما به test اضافه کردیم را باید کم کنیم.

فاز دوم

قبل از ورود به بخش ها ابتدا کمی با کتابخانه و توابع آشنا شویم. ما در این پروژه از کتابخانه cryptography استفاده میکنیم. همچنین ما از رمزگذاری متقارن استفاده میکنیم. این کتاب خانه بر روی الگوریتم AES بنا شده است.

در ابتدا یک فایل keyGenerator.py ایجاد می کنیم. در این فایل قطعه کد زیر قرار داده شده است:

```
from cryptography.fernet import Fernet  
  
def write_key():  
    """  
    Generates a key and save it into a file  
    """  
    key = Fernet.generate_key()  
    with open("key1.key", "wb") as key_file:  
        key_file.write(key)  
  
write_key()
```

Fernet از الگوریتم همگام سازی متقابل (symmetric encryption) استفاده می کند و برای رمزنگاری اطلاعات از (AES) با استفاده از حالت (CBC) استفاده می کند. همچنین از HMAC برای امضاء داده ها استفاده می کند تا اطمینان حاصل شود که داده ها در حین انتقال تغییر نکرده اند.

در کد بالا یک کلید برای رمزنگاری و رمزگشایی تولید میشود و آن را در یک فایل به نام key.key قرار داده و آن را در مسیر پروژه ذخیره می کنیم.

بخش اول:

در بخش اول رمز نگاری فایل انجام می شود. کد آن به صورت زیر است:

```
from cryptography.fernet import Fernet

def load_key():
    """
    Loads the key from the current directory named `key.key`
    """
    return open("key.key", "rb").read()

def encrypt(filename, key):
    """
    Given a filename (str) and key (bytes), it encrypts the file and write it
    """
    f = Fernet(key)
    with open(filename, "rb") as file:
        # read all file data
        file_data = file.read()
    # encrypt data
    encrypted_data = f.encrypt(file_data)
    # write the encrypted file
    with open(filename, "wb") as file:
        file.write(encrypted_data)
```



```
key = load_key()
encrypt("file.txt", key=key)
```

این بخش دو تابع دارد. یکی برای لود کردن کلید و دیگری برای encrypt فایل. برای لود کردن فایل را میخواند و محتویات آن را بر میگرداند.

در تابع دوم که برای رمزگذاری است مقدار کلید و اسم فایل که می‌خواهیم رمز گذاری روی آن انجام شود، به عنوان ورودی داده می‌شود.

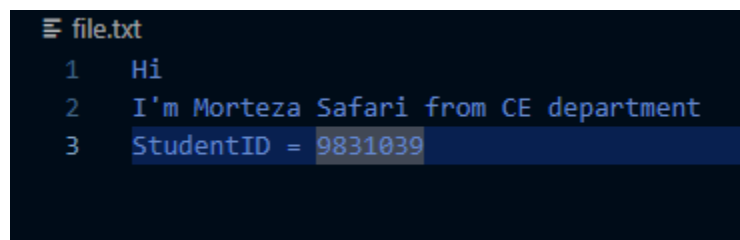
```
f = Fernet(key)
```

در اینجا یک شی از کلیدی که داریم ایجاد می‌شود.

در ادامه فایل را خوانده و با استفاده از کد زیر رمزنگاری صورت می‌گیرد. همانطور که گفته شد رمزنگاری با استفاده از الگوریتم AES انجام می‌شود.

```
encrypted_data = f.encrypt(file_data)
```

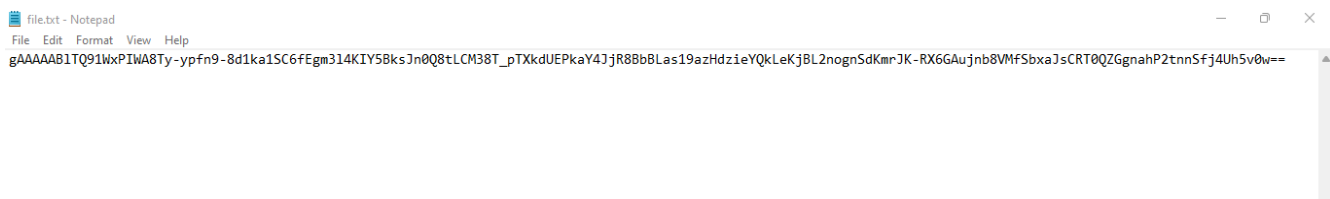
سپس فایل رمزنگاری شده با همان نام قبلی ذخیره می‌شود.



```
file.txt
1 Hi
2 I'm Morteza Safari from CE department
3 StudentID = 9831039
```

plaintext

پس از رمز نگاری به صورت زیر می‌شود.



```
file.txt - Notepad
File Edit Format View Help
gAAAAAB1TQ91WxPIWA8Ty-ypfn9-8d1ka1SC6fEgm314KIY5BksJn0Q8tLCM38T_pTXkdUEPkaY4JjR88bBLas19azHdzieYQkLeKjBL2nognSdKmrJK-RX6GAujnb8VMf5bxaJsCRT0QZGgnahP2tnnSfj4Uh5v0w==
```

Ciphertext

بخش دوم

در این بخش عملیات رمزگشایی فایل انجام می‌شود. کد آن به صورت زیر است:

```
from cryptography.fernet import Fernet

def load_key():
    """
    Loads the key from the current directory named `key.key`
    """
    return open("key.key", "rb").read()

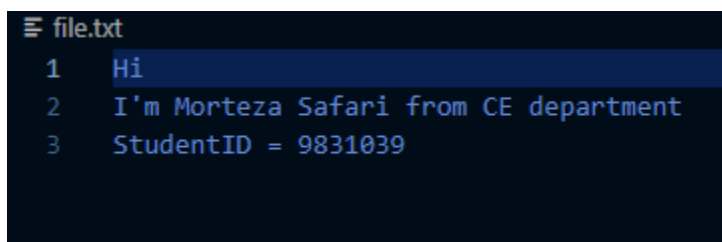
def decrypt(filename, key):
    """
    Given a filename (str) and key (bytes), it decrypts the file and write it
    """
    f = Fernet(key)
    with open(filename, "rb") as file:
        # read the encrypted data
        encrypted_data = file.read()
    # decrypt data
    decrypted_data = f.decrypt(encrypted_data)
    # write the original file
    with open(filename, "wb") as file:
        file.write(decrypted_data)

key = load_key()
decrypt("file.txt", key=key)
```

همانطور که ملاحظه میکنید کد این بخش همانند کد بخش قبل است با این تفاوت که در اینجا از تابع `decrypt` استفاده می‌شود.

```
decrypted_data = f.decrypt(encrypted_data)
```

پس از رمزگشایی عبارت رمز شده که نشان داده شد خروجی به صورت زیر است:

A screenshot of a text editor window titled 'file.txt'. The window contains three lines of text: '1 Hi', '2 I'm Morteza Safari from CE department', and '3 StudentID = 9831039'. The first line is highlighted with a blue background.

```
file.txt
1 Hi
2 I'm Morteza Safari from CE department
3 StudentID = 9831039
```

تصویر پس از رمزگشایی

باتشکر از توجه شما