

دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)

گزارش پروژه ۳ درس امنیت اطلاعات

دانشجو:

مرتضی صفری (۹۸۳۱۰۳۹)

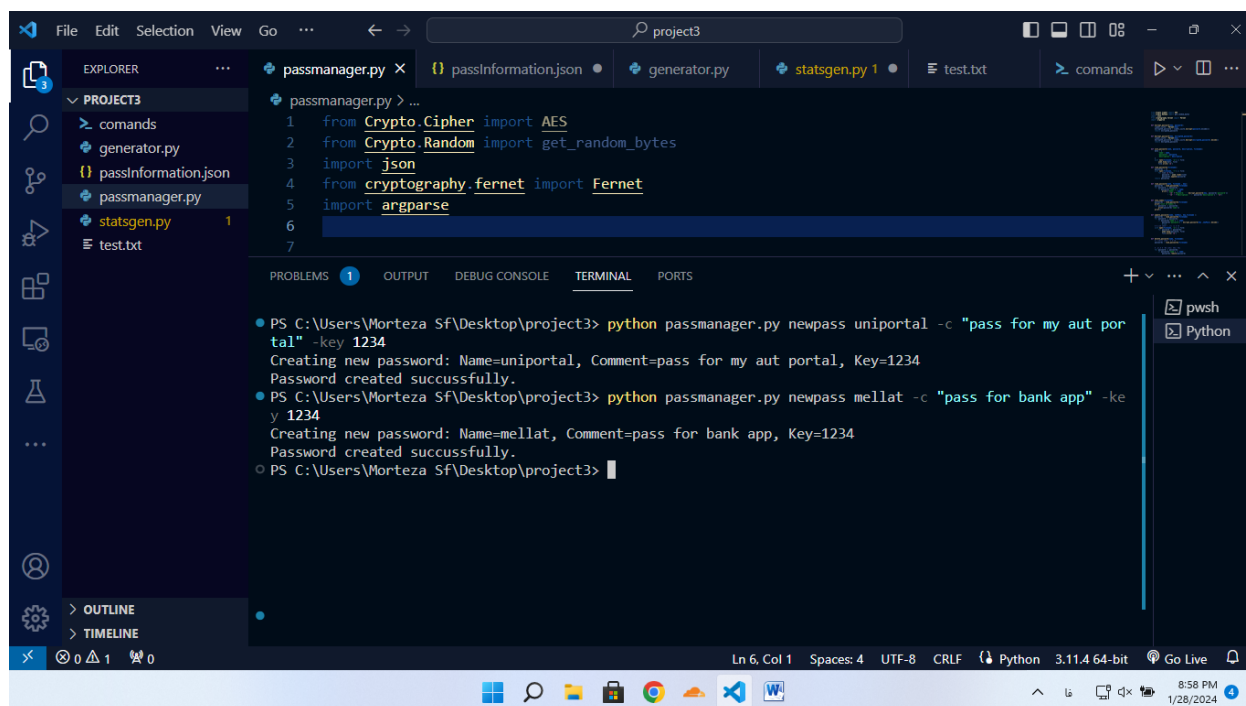
استاد: دکتر شهریاری

بهمن ماه ۱۴۰۲

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

خروجی بخش اول:

ساخت رمز و مشخصات:



The screenshot displays the Visual Studio Code interface with a project named 'project3'. The Explorer panel on the left shows the file structure, including 'comands', 'generator.py', 'passInformation.json', 'passmanager.py', 'statsgen.py', and 'test.txt'. The main editor window shows the code for 'passmanager.py', which imports AES, get_random_bytes, json, Fernet, and argparse. The terminal at the bottom shows the execution of the script with two commands: 'python passmanager.py newpass uniportal -c "pass for my aut portal" -key 1234' and 'python passmanager.py newpass mellat -c "pass for bank app" -key 1234'. Both commands result in successful password creation.

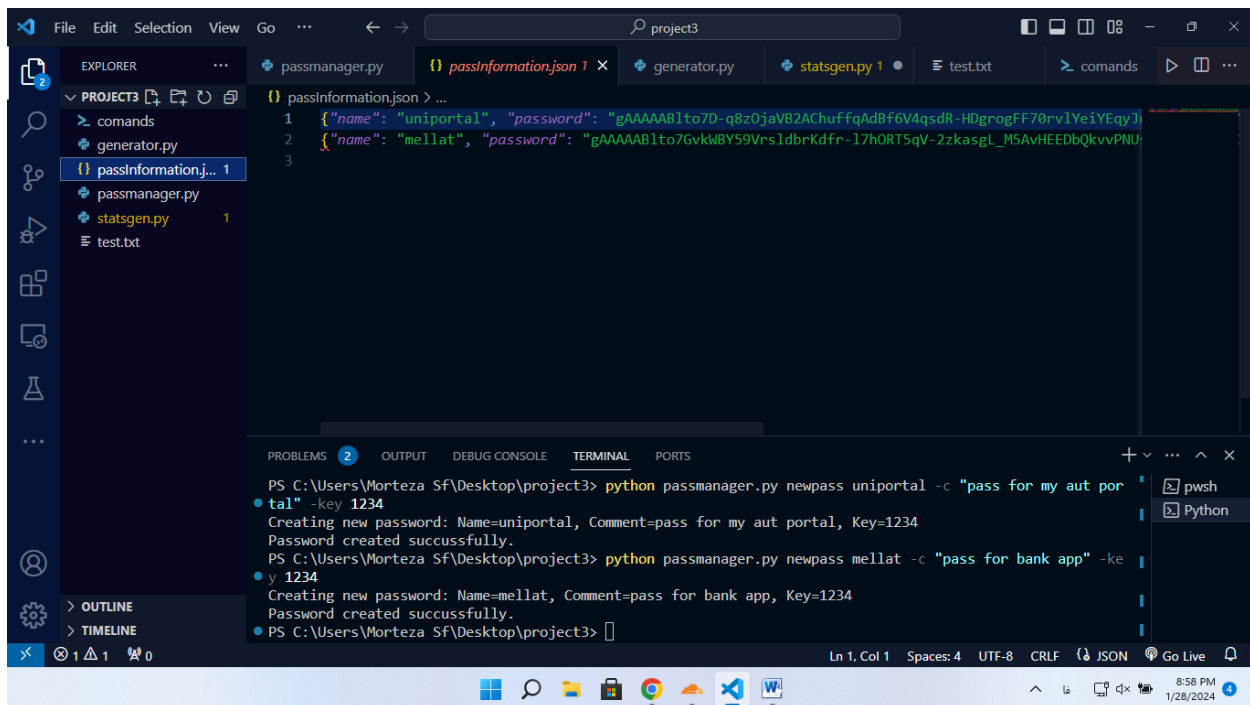
```
1 from Crypto.Cipher import AES
2 from Crypto.Random import get_random_bytes
3 import json
4 from cryptography.fernet import Fernet
5 import argparse
6
7
```

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS

- PS C:\Users\Morteza Sf\Desktop\project3> python passmanager.py newpass uniportal -c "pass for my aut portal" -key 1234
Creating new password: Name=uniportal, Comment=pass for my aut portal, Key=1234
Password created succussfully.
- PS C:\Users\Morteza Sf\Desktop\project3> python passmanager.py newpass mellat -c "pass for bank app" -key 1234
Creating new password: Name=mellat, Comment=pass for bank app, Key=1234
Password created succussfully.
- PS C:\Users\Morteza Sf\Desktop\project3>

Ln 6, Col 1 Spaces: 4 UTF-8 CRLF Python 3.11.4 64-bit Go Live

نمایش:



ذخیره در یک فایل json

```
{
  "name": "uniportal",
  "password": "gAAAAAB1to7D-q8z0jaVB2AChuffqAdBf6V4qsdR-HDgrogFF70rv1YeiYEgyJ",
  "description": "pass for my aut portal"
},
{
  "name": "mellat",
  "password": "gAAAAAB1to7GvkWBY59Vrs1dbrKdfr-17h0RT5qV-2zkasgL_M5AvHEEDbQkvvPNU",
  "description": "pass for bank app"
}
```

نمایش و آپدیت رمز خاص:

The screenshot shows a VS Code window with a terminal at the bottom. The terminal output shows the following sequence of commands and results:

```
1 from Crypto.Cipher import AES

Password created succussfully.
PS C:\Users\Morteza Sf\Desktop\project3> python passmanager.py sel mellat

Showing password value and comment for: Name=mellat
Name:mellat
Password: 1234
Description: pass for bank app

PS C:\Users\Morteza Sf\Desktop\project3> python passmanager.py update mellat -key 4321

Updating password value for: Name=mellat

PS C:\Users\Morteza Sf\Desktop\project3> python passmanager.py sel mellat

Showing password value and comment for: Name=mellat
Name:mellat
Password: 4321
Description: pass for bank app

PS C:\Users\Morteza Sf\Desktop\project3>
```

The Explorer sidebar on the left shows the file structure of the project, including `passmanager.py`, `passInformation.json`, `generator.py`, `statsgen.py`, `test.txt`, and `comands`.

حذف رمز خاص:

The screenshot shows the VS Code interface with the `passInformation.json` file open in the editor. The JSON content is as follows:

```
{
  "name": "uniportal",
  "password": "gAAAAAB1to7D-q8z0jaVB2AChuffqAdBf6V4qsdR-HDgrogFF70rv1YeiYEgyJ"
}
```

The terminal at the bottom shows the command to delete the password entry:

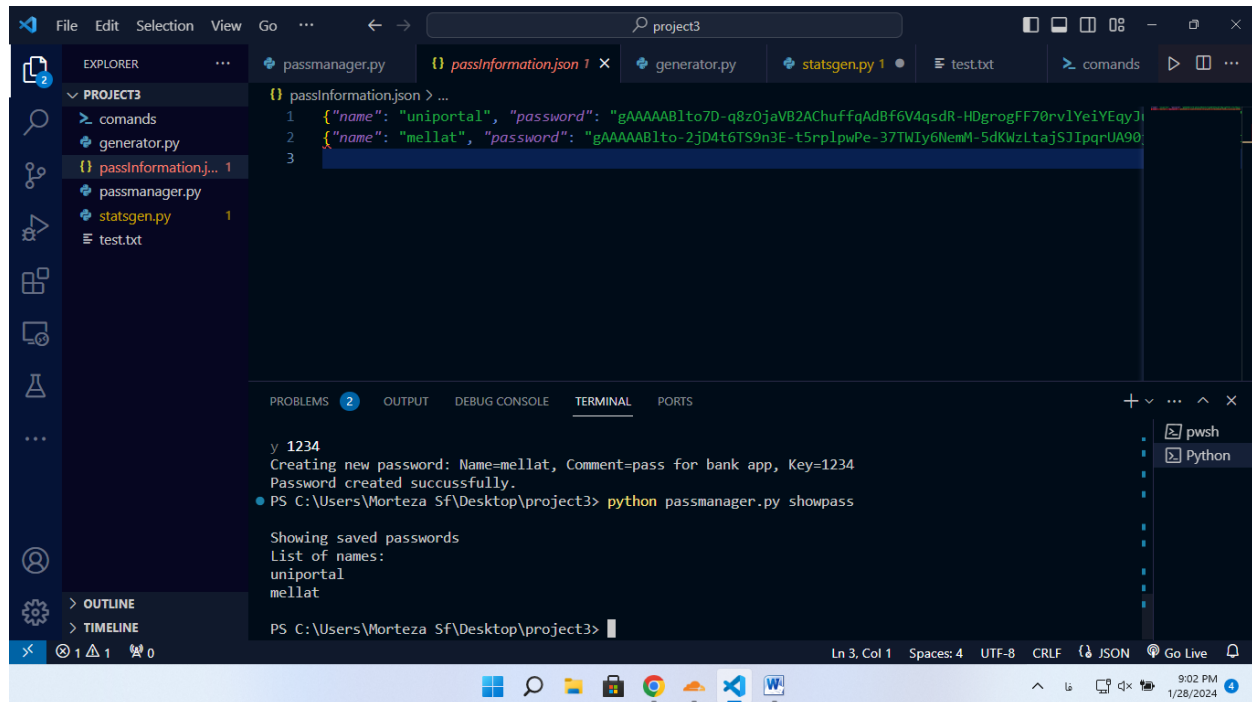
```
PS C:\Users\Morteza Sf\Desktop\project3> python passmanager.py del mellat

Deleting password: Name=mellat

PS C:\Users\Morteza Sf\Desktop\project3>
```

The Explorer sidebar on the left shows the file structure, including `passInformation.json`.

نمایش همه رمز ها:



The screenshot shows a Visual Studio Code editor window with a project named 'project3'. The Explorer sidebar on the left shows the file structure: 'commands', 'generator.py', 'passInformation.json', 'passmanager.py', 'statsgen.py', and 'test.txt'. The main editor area displays the 'passInformation.json' file, which contains two entries:

```
{  
  "name": "uniportal", "password": "gAAAAAB1to7D-q8z0jaVB2AChuFfqAdBf6V4qsdR-HDgrogFF70rv1YeiYEgyJi",  
  "name": "mellat", "password": "gAAAAAB1to-2jD4t6TS9n3E-t5rplpwPe-37TWIy6NemM-5dKWzLtajSJIpqrUA90"  
}
```

Below the editor, the TERMINAL panel is active, showing the output of a Python script. The terminal text is as follows:

```
y 1234  
Creating new password: Name=mellat, Comment=pass for bank app, Key=1234  
Password created succussfully.  
PS C:\Users\Morteza Sf\Desktop\project3> python passmanager.py showpass  
  
Showing saved passwords  
List of names:  
uniportal  
mellat  
  
PS C:\Users\Morteza Sf\Desktop\project3>
```

The status bar at the bottom indicates the current cursor position is 'Ln 3, Col 1' and the file encoding is 'UTF-8'.

خروجی بخش دوم:

در این بخش نمیشد از الگوریتم رمزنگاری بخش یک استفاده کرد زیرا طول رمز تولید شده به طور تصاعدی بالا می‌رفت. بنابراین ارز تابع هش md5 به طور متوالی استفاده شد.

بخشی از فایل:

```
test.txt - Notepad
File Edit Format View Help
9af15b336e6a9619928537df30b2e6a2376569fcf9d7e773eccede65606529a0
c5e3cb980ee22a5b7150e88263efe6a13baa54e98a0cfd8d22b751c1679aacc
2aed786fb0641cb3414971c6ca52c452265c8a0dab3853175b1127ce040022e
e8167ff9ba93ee511c4832e3d717ec452462ee6f540629d48e66d1d98db23184
3dea33191fb6d6843f54f3df0e7f5518ac7198bc6e6f47211f90f801e780505a
f944d874b1cc3f3450b52d6486c13f5855a1e6dc2dcfef47cce618a89d887548
a08a182c72d670d3992453bd7cd95343766f2bce25225aba495bce5066984f01
d67131e77a0a6a049fc68f9a5ce370e6bd11df6230edd5c70424db6199556ee0
0a283d32c44685a0381ea3936395f7ebe30cf6650fb3f4ab8ca4da34e8dfb0e8
b2447ed6a968277211b46c423fc4d68a28c4505a6663f16cf483652d7f633573
9cfa3b20e4a225189c67d8646b628e77da3ead15739d3f74ab80127b88c3e1ceb
096e50119c7a8585d5424378b3a84673a564eb4d8a1904d28aacad42126978ae
3917b9f6b3d8038e09442c9994f40ea1abafef9fb2ca8d99e0f803e4e6cada4a
eca55f343faa16fcd543772330542518d5bb57c53dde214ff16d801bf3e323c4
977657f43584820c446c92c9946bc0b1a8fe427ee35046d6120107123c9106cc
7d77f819e36019c5525d313034d18730aed4a78989761b6e0fef35d78c0d113d8
fb8aee88160a78738539bdc81f52f02aed332dd0b1294c4a4817ec34c718f8e
7c3daf9c3eeaa169b7207d98f83b276793a6056c23f44e4185a833479a74c94f
588e18ead075bb5dc01f431e3fd75e89d507a5154cabfb191b08b449682e240d
1ccc4ee631b3e5d05ae67a3ab1303117fb98d6c92ee16dc0ef7ed8c95300a985
514bc41204cdaaf2e7c9c8030bb2139fc0f1502079e130a9c9b98456e78dd23e
eebfd3976c70798458840128c8d297257a2385bfc103cbc64c6945395785676d
35192730a18dedeb3a5a741b04652d3d8a7534c68cd0bb66ba4f137d4fe06d82
2aafd89752e6f076eab8f44959b02bf9aa92d0869df82207812f80188ab2706e
8f6d1259ebbd273d02bde80dd00bf4d96b574f3fd9421db813bc4944c54c66c6
63a9292f7f3d739723565979dc3984c25fb0e2e8575acd07c0db1db3250191a1
223089189716b82b70d4be9fb94f0fb7ea5eaf2ffcf0f13bcb489d13d1f24ab0
402ef52fabe642eabe4f13c83d73dc27a98e1c9cb41d3d8ceb5381e6a9ad35cc
236c628b40e48a37e1555a015ca059728865c1d38ef78f857e7ec05ca48d43dc
d6160f9f4d5bdf3078729bb0785bc0fd8a00c74baabf9b83665ce83cd5d268c8
4534a31e16bdc5ccd2df30b94a218e94d8a24e590a53982a257dd3a7ed2272f6
16507e1be01963cddcf651d77384a9d97df0bf819b6f59da83d734e640b84f1ac
20423d64468240388c4672867a7cc63c096919f23a3eb1de260b1e2b5f1773ab
4af019dec11ee1fc6aacfe500b089b48b3a77bb1d444a98d9795c93a8125c799
bae1c1d454e56b8fb546936801c46df0bd259dfa8f5a4d965e3a8a8d81f5097f
File Edit Format View Help
```

آزمایش ۱۰ هزار رمز فایل بالا:

The screenshot shows the Visual Studio Code (VS Code) editor interface. The top menu bar includes File, Edit, Selection, View, Go, and a search icon. The left sidebar contains icons for Explorer, Search, Source Control, Run and Debug, Extensions, and Settings. The main editor area displays a Python file named `statsgen.py` with the following code:

```
201 print("\n[*] Advanced Masks:")
202 for (advancedmask,count) in sorted(iter(self.stats_advancedmasks.items()), key=operator.itemgetter(1), reverse=True):
203     if count*100/self.filter_counter > 0:
```

The bottom panel shows the TERMINAL view with the command prompt `PS C:\Users\Morteza SF\Desktop\project3> python statsgen.py test.txt`. The status bar at the bottom indicates the current line and column (Ln 212, Col 53), the number of spaces (4), the encoding (UTF-8), the line feed (LF), the Python version (3.11.4 64-bit), and the Go Live status.

[illegible]

توضیح کد:

بخش اول:

کتابخانه های لازم:

```
from Crypto.Cipher import AES
from Crypto.Random import get_random_bytes
import json
from cryptography.fernet import Fernet
import argparse
```

توابع رمزگذاری و رمزگشایی بر مبنای AES :

```
def encrypt_password(key, password):
    cipher_suite = Fernet(key)
    encrypted_password = cipher_suite.encrypt(password.encode())
    return encrypted_password

def decrypt_password(key, encrypted_password):
    cipher_suite = Fernet(key)
    decrypted_password = cipher_suite.decrypt(encrypted_password).decode()
    return decrypted_password
```

ذخیره اطلاعات در قالب یک فایل json:

```
def save_password(name, password, description, filename):
    data = {
        'name': name,
        'password': password,
        'description': description
    }
    with open(filename, 'a') as file:
        json.dump(data, file)
        file.write('\n')
```

تابع زیر برای برگرداندن لیست رمز هاست:

```
def load_passwords(filename):  
    passwords = []  
    with open(filename, 'r') as file:  
        for line in file:  
            password = json.loads(line)  
            passwords.append(password)  
    return passwords
```

تابع زیر برای نمایش اطلاعات یک رمز خاص می باشد:

```
def load_password(name, filename , key):  
    passwords = load_passwords(filename)  
    for password in passwords :  
        if password['name'] == name:  
            print(f'Name: ' + name  
                  + "\n" f'Password: ' + decrypt_password(key,  
password['password']))  
            + "\n" + f'Description: ' + password['description'] + "\n")
```

نمایش اسم پسورد ها:

```
def show_names(filename):  
    passwords = load_passwords(filename)  
    print("List of names:")  
    for password in passwords:  
        print(password['name'])  
    print()
```

آپدیت پسورد ها:

```
def update_password(name, newPass, key,filename ):
```

```

passwords = load_passwords(filename)
for password in passwords:
    if password['name'] == name:
        password['password'] = encrypt_password(key, newPass).decode()
        break
# ذخیره رمزهای جدید در فایل
with open(filename, 'w') as file:
    for password in passwords:
        json.dump(password, file)
        file.write('\n')

```

حذف یک پسورد خاص با استفاده از نام آن:

```

def delete_password(name, filename):
    # بارگیری رمزها از فایل
    passwords = load_passwords(filename)

    # جستجو و حذف رکورد مورد نظر
    for password in passwords:
        if password['name'] == name:
            passwords.remove(password)
            break

    # ذخیره رمزهای جدید در فایل
    with open(filename, 'w') as file:
        for password in passwords:
            json.dump(password, file)
            file.write('\n')

```

قطعه زیر کد یک تابع به نام `create_argument_parser` را تعریف می‌کند که یک پارسر آرگومان (Argument Parser) برای مدیریت دستورات برنامه رمزعبور ایجاد می‌کند. این پارسر تنظیمات مربوط به دستورات مختلف را تعریف می‌کند.

```

def create_argument_parser():
    parser = argparse.ArgumentParser(prog='passmanager.py', description='Password
Manager')

    subparsers = parser.add_subparsers(dest='command', title='Commands')

    # Command: newpass
    newpass_parser = subparsers.add_parser('newpass', help='Create a new
password')
    newpass_parser.add_argument('name', help='Name of the password')
    newpass_parser.add_argument('-c', '--comment', help='Comment for the
password')
    newpass_parser.add_argument('-key', '--password', help='User simple
password')

    # Command: --showpass
    subparsers.add_parser('showpass', help='Show saved passwords')

    # Command: --sel
    sel_parser = subparsers.add_parser('sel', help='Show password value and
comment')
    sel_parser.add_argument('name', help='Name of the password')

    # Command: --update
    update_parser = subparsers.add_parser('update', help='Update password value')
    update_parser.add_argument('name', help='Name of the password')
    update_parser.add_argument('-key', '--password', help='User new password')

    # Command: --del
    del_parser = subparsers.add_parser('del', help='Delete a password')
    del_parser.add_argument('name', help='Name of the password')

    return parser

```

در ادامه تابع main برنامه را داریم:

در این تابع بررسی دستورات وارد شده پرداخته میشود و عملیات های لازم انجام می شود.

```

def main():

    key = b't_-wUtK2eEXoPHyQxRRL_x0typzZi1IQhzk_GxLsH_E='

```

```

filename = "passInformation.json"

parser = create_argument_parser()
args = parser.parse_args()

if args.command == 'newpass':
    # Handle newpass command
    print(f'Creating new password: Name={args.name}, Comment={args.comment},
Key={args.password}')
    cipherText = encrypt_password(key ,args.password).decode()
    save_password(args.name, cipherText , args.comment, filename)
    print("Password created succussfully.")

elif args.command == 'showpass':
    # Handle showpass command
    print()
    print('Showing saved passwords')
    show_names(filename)

elif args.command == 'sel':
    # Handle sel command
    print()
    print(f'Showing password value and comment for: Name={args.name}')
    load_password(args.name,filename,key)

elif args.command == 'update':
    # Handle update command
    print()
    print(f'Updating password value for: Name={args.name}')
    update_password(args.name, args.password , key , filename= filename)
    print()

elif args.command == 'del':
    # Handle del command
    print()
    print(f'Deleting password: Name={args.name}')
    delete_password(args.name, filename)
    print()

```

نمونه دستورات به شرح زیر است:

اضافه کردن رمز:

```
python passmanager.py newpass mellat -c "pass for bank app" -key 1234
```

نمایش اطلاعات یک رمز خاص با استفاده از نام آن:

```
python passmanager.py sel mellat
```

بروزرسانی و تغییر یک رمز خاص:

```
python passmanager.py update mellat -key 4321
```

حذف یک رمز خاص با استفاده از نام آن:

```
python passmanager.py del mellat
```

نمایش تمامی اسامی رمز های کاربر:

```
python passmanager.py showpass
```

تمام کد در ادامه برای ساخت ۱۰۰۰۰ رمز برای تست و بررسی آورده شده ایست:

```
import hashlib

from cryptography.fernet import Fernet

def encrypt_password(key, password):
    cipher_suite = Fernet(key)
    encrypted_password = cipher_suite.encrypt(password.encode())
    return encrypted_password

def constant_hash(input_string):
    md5_hash = hashlib.md5(input_string.encode()).hexdigest()

    return md5_hash

def main():

    key = b't_-wUtK2eEXoPHyQxRRL_x0typzZi1IQhzk_GxLsH_E='
    text = "0000"

    for i in range(10000):
        print(i)

        text = constant_hash(text)
        with open('test.txt', 'a') as file:
            file.write(text + "\n")

if __name__ == "__main__":
    main()
```

توجه: همانطور که ملاحظه می‌فرمایید در ابتدا قصد استفاده از همان روش رمزنگاری بخش اول را داشتم اما روند تولید رمز به شکل صعودی طول رمزها را تغییر میدهد. به همین دلیل از md5 برای هش کردن رمز استفاده شد. و ۱۰۰۰۰ رمز را خط به خط در یک فایل test.txt ذخیره کردیم و سپس توسط statsgen.py مورد ارزیابی قرار دادیم.

باغ تشکر از توجه شما

