1. مدیریت گذرواژه

1.1 تولید گذرواژه

رویکرد ما به تولید گذرواژه اولویت امنیت را دارد. ما از هش SHA-512 با سالت برای یکتایی و انکدینگ base64 برای افزایش امنیت و ذخیره‌سازی استفاده می‌کنیم. قطعه کد زیر تابع تولید گذرواژه ما را نشان می‌دهد.

```
def generate_secure_password(user_input, password_length):
    salt = os.urandom(16)
    combined_input = salt + user_input.encode()
    hashed = hashlib.sha512(combined_input).digest()
    return b64encode(hashed)[:password_length].decode('utf-8')
```

1.2 رمزنگاری گذرواژه

ما از رمزنگاری AES با حالت Cipher Block Chaining (CBC) برای حفظ محرمانه ویژگی گذرواژه استفاده می‌کنیم. قطعه کد نشان‌دهنده رمزنگاری AES با برداشت اولیه (IV) است.

```
def encrypt_text(data, encryption_key):
    cipher = AES.new(pad(encryption_key.encode(), AES.block_size), AES.MODE_CBC)
    ciphertext_bytes = cipher.encrypt(pad(data.encode(), AES.block_size))
    return b64encode(cipher.iv + ciphertext_bytes).decode('utf-8')
```

1.3 ذخیره‌سازی و مدیریت گذرواژه

توابع ما برای ذخیره‌سازی امن و مدیریت، سریالایز کردن داده گذرواژه به فرمت JSON، رمزنگاری آن و نوشتن در فایل را شامل می‌شود. تابع بارگذاری داده‌ها را باز می‌کند و آنها را بارگذاری می‌کند.

```
def load_encrypted_data(decryption_key):
    if not os.path.exists('passwords.enc'):
        return {}
    with open('passwords.enc', 'r') as file:
        encrypted_data = file.read()
    data = decrypt_text(encrypted_data, decryption_key)
    return json.loads(data)


def save_encrypted_data(data, encryption_key):
    data_to_save = json.dumps(data)
    encrypted_data = encrypt_text(data_to_save, encryption_key)
    with open('passwords.enc', 'w') as file:
        file.write(encrypted_data)
```

1.4 احراز هویت کاربر

احراز هویت کاربر امانت گذرواژه‌های ذخیره شده را تضمین می‌کند. تابع `is_valid_verification_key` صحت کلید رمزنگاری را بررسی می‌کند.

```
def is_valid_encryption_key(encryption_key):
    if os.path.exists('passwords.enc'):
        try:
            load_encrypted_data(encryption_key)
            return True
        except ValueError:
            return False
    return True
```

2. StatsGen - ابزار تجزیه و تحلیل آماری گذرواژه

***به دلیل عدم پشتیبانی پایتون 2، کد تغییر داده شد.**

2.1 تحلیل گذرواژه

StatsGen گذرواژه‌ها را به ماسک‌های ساده و پیشرفته دسته‌بندی می‌کند و برچسب‌هایی در مورد ویژگی‌های گذرواژه ارائه می‌دهد.

```
def analyze_password(self, password):
    # Password length
    pass_length = len(password)

    # Character-set and policy counters
    digit = 0
    lower = 0
    upper = 0
    special = 0

    simplemask = list()
    advancedmask_string = ""
```

2.2 معیارهای آماری

StatsGen آمار جامعی در مورد ویژگی‌های گذرواژه از جمله توزیع طول، نسبت‌های مجموعه نویسه‌ها و معیارهای پیچیدگی گذرواژه تولید می‌کند.

```
def generate_stats(self, filename):
    """ Generate password statistics. """

    with open(filename, 'r') as f:
        for password in f:
            password = password.rstrip('\r\n')

            if len(password) == 0:
                continue

            self.total_counter += 1
```

```
def print_stats(self):
    """ Print password statistics. """

    print("[+] Analyzing %d%% (%d/%d) of passwords" % (
        self.filter_counter * 100 / self.total_counter, self.filter_counter, self.total_counter))
    print("    NOTE: Statistics below is relative to the number of analyzed passwords, not total number of passwords")
    print("\n[*] Length:")
    for (length, count) in sorted(self.stats_length.items(), key=operator.itemgetter(1), reverse=True):
        if self.hiderare and not count * 100 / self.filter_counter > 0:
            continue
        print("[+] %25d: %02d%% (%d)" % (length, count * 100 / self.filter_counter, count))

    print("\n[*] Character-set:")
    for (char, count) in sorted(self.stats_charactersets.items(), key=operator.itemgetter(1), reverse=True):
        if self.hiderare and not count * 100 / self.filter_counter > 0:
            continue
        print("[+] %25s: %02d%% (%d)" % (char, count * 100 / self.filter_counter, count))

    print("\n[*] Password complexity:")
    print("[+]                    digit: min(%s) max(%s)" % (self.mindigit, self.maxdigit))
    print("[+]                    lower: min(%s) max(%s)" % (self.minlower, self.maxlower))
    print("[+]                    upper: min(%s) max(%s)" % (self.minupper, self.maxupper))
    print("[+]                  special: min(%s) max(%s)" % (self.minspecial, self.maxspecial))
```

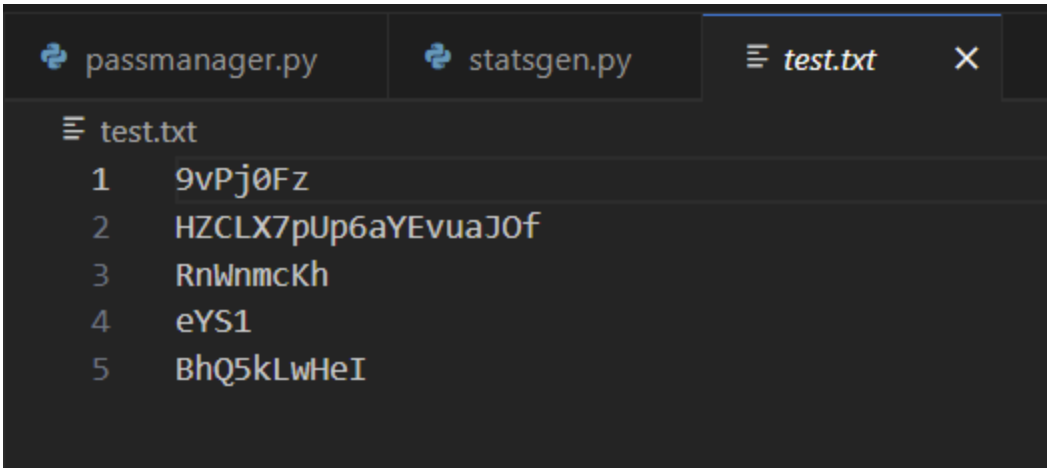
2.3 فیلتر و خروجی

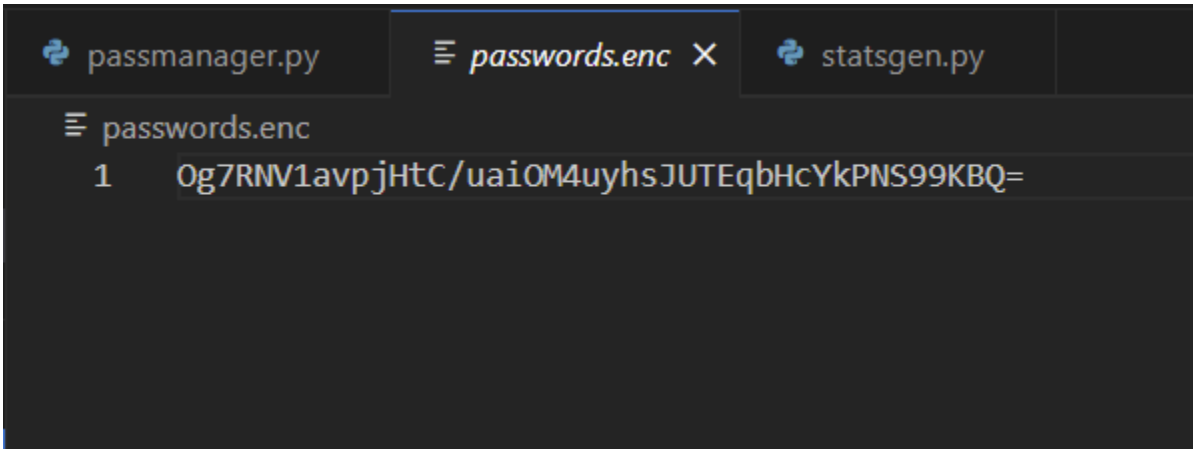
StatsGen به کاربران این امکان را می‌دهد که گذرواژه‌ها را بر اساس طول و مجموعه نویسه فیلتر کنند و امکانات تحلیل را فراهم می‌کند.

3. خروجی ها

```
PS C:\Users\Reza\vscode\securityHW3> python passmanager.py create myportal -comment "Password for university portal" -encryption_key 1234 -password_length 16
>> python passmanager.py display -encryption_key 1234
>> python passmanager.py select myportal -encryption_key 1234
>> python passmanager.py update myportal -encryption_key 1234 -password_length 20
>> python passmanager.py display -encryption_key 1234
>> python passmanager.py delete myportal -encryption_key 1234
>> python passmanager.py generate -number 5
>> python statsgen.py test.txt
Password entry for myportal created.
Name: myportal    Password: edyaPj8onDVayyN6    Comment: Password for university portal

Password: edyaPj8onDVayyN6    Comment: Password for university portal
Password entry for myportal updated.
Name: myportal    Password: bavGgLL+Qg/0PQ7kSWNH    Comment: Password for university portal

Password entry for myportal deleted.

     StatsGen 0.0.3   | |
     _ __   __ _ ___| | _
    | '_ \ / _` |/ __| |/ /
    | |_) | (_| | (__|   <
    | .__/ \__,_|\___|_|\_\
    | |
    |_| iphelix@thesprawl.org


[*] Analyzing passwords in [test.txt]
[+] Analyzing 100% (5/5) of passwords
    NOTE: Statistics below is relative to the number of analyzed passwords, not total number of passwords

[*] Length:
[+]                         7: 20% (1)
[+]                        19: 20% (1)
[+]                         8: 20% (1)
[+]                         4: 20% (1)
[+]                        10: 20% (1)

[*] Character-set:
[+]              mixedalphanum: 80% (4)
[+]                 mixedalpha: 20% (1)

[*] Password complexity:
[+]                      digit: min(0) max(2)
[+]                      lower: min(1) max(7)
[+]                      upper: min(2) max(10)
[+]                    special: min(0) max(0)
```

```
[*] Simple Masks:
[+]                 othermask: 40% (2)
[+]                    string: 20% (1)
[+]               stringdigit: 20% (1)
[+]          stringdigitstring: 20% (1)

[*] Advanced Masks:
[+]              ?d?l?u?l?d?u?l: 20% (1)
[+] ?u?u?u?u?u?d?l?u?l?d?l?u?u?l?l?l?u?u?l: 20% (1)
[+]           ?u?l?u?l?l?l?u?l: 20% (1)
[+]                  ?l?u?u?d: 20% (1)
[+]       ?u?l?u?d?l?u?l?u?l?u: 20% (1)
PS C:\Users\Reza\vscode\securityHW3>
```


passmanager.py
statsgen.py
test.txt
test.txt
1 9vPj0Fz
2 HZCLX7pUp6aYEvuaJOf
3 RnWnmcKh
4 eYS1
5 BhQ5kLwHeI


passmanager.py
passwords.enc
statsgen.py
passwords.enc
1 Og7RNV1avpjHtC/uaiOM4uyhsJUTEqbHcYkPNS99KBQ=