Міністерство освіти і науки України

Національний технічний університет «ХПІ»

Навчально-науковий інститут комп’ютерних наук та інформаційних технологій

Кафедра комп’ютерної інженерії та програмування

# ЗВІТ

з лабораторної роботи № 10

з дисципліни «Сучасні технології безпечного програмування»

**«СТВОРЕННЯ ЛІЦЕНЗІЙНОГО КЛЮЧА»**

Виконав:

студент гр. КН-Н922б

Кулик Д.І.

Перевірив:

Бульба С. С.

Харків – 2022

**Мета роботи:** дослідити і порівняти існуючі механізми створення і перевірки валідності ліцензійних ключів.

**Індивідуальне завдання**

Дослідити існуючі механізми створення і перевірки валідності ліцензійних ключів. Зробити порівняльну характеристику кожного механізму.

Реалізувати один з методів генерації (та читання/верифікації) ліцензійного ключа. Довести доцільність обраного методу.

## Хід роботи

Реалізуємо кілька механізмів створення і перевірки валідності ліцензійних ключів: за допомогою RSA та SHA1.

**Важливі фрагменти програми**

```
class RSAKeyGen:

    def __init__(self, path_to_public_key, path_to_private_key):
        self.path_to_public_key = path_to_public_key
        self.path_to_private_key = path_to_private_key

    def generate_license(self, email):
        with open(self.path_to_private_key, 'rb') as file:
            key = rsa.PrivateKey.load_pkcs1(file.read())
        return base64.b64encode(rsa.sign(email.encode(), key, 'SHA-1')).decode()

    def new_rsa(self):
        public, private = rsa.newkeys(512)
        with open(self.path_to_public_key, 'wb') as file:
            file.write(public.save_pkcs1())
        with open(self.path_to_private_key, 'wb') as file:
            file.write(private.save_pkcs1())

    def valid(self, email, license_key):
        with open(self.path_to_public_key, 'rb') as file:
            key = rsa.PublicKey.load_pkcs1(file.read())
        try:
            rsa.verify(email.encode(), base64.b64decode(license_key), key)
        except rsa.VerificationError:
            return False
        else:
            return True
```

**Рисунок 1 –** Механізм генерації ключів за допомогою RSA

```python
class SHA1KeyGen:

    def __init__(self, secret):
        self.secret = secret.encode()

    def generate_license(self, email):
        hashed = hmac.new(self.secret, email.encode(), hashlib.sha1)
        return base64.encodebytes(hashed.digest()).decode('utf-8').rstrip('\n')

    def valid(self, email, key):
        hashed = hmac.new(self.secret, email.encode(), hashlib.sha1)
        return base64.encodebytes(hashed.digest()).decode('utf-8').rstrip('\n') == key
```

**Рисунок 2 –** Механізм генерації ключів за допомогою SHA1

```python
if __name__ == '__main__':
    email = 'daniil2022kulyk@mail.com'

    PUBLIC_KEY_PATH = './public_key.pem'
    PRIVATE_KEY_PATH = './private_key.pem'
    keygen = RSAKeyGen(PUBLIC_KEY_PATH, PRIVATE_KEY_PATH)
    keygen.new_rsa()
    key = keygen.generate_license(email)
    print(f'RSA: LICENSE KEY   : {key}')
    is_valid = keygen.valid(email, key)
    print(f'RSA: IS VALID      : {is_valid}')

    SECRET = 'fa4db30478c45ef'
    keygen = SHA1KeyGen(SECRET)
    key = keygen.generate_license(email)
    print(f'SHA1: LICENSE KEY   : {key}')
    is_valid = keygen.valid(email, key)
    print(f'SHA1: IS VALID      : {is_valid}')
```

**Рисунок 3 –** Запуск обох механізмів генерацій ключів

## Результати роботи програми

```
C:\Users\Daniil\PycharmProjects\stbp\Scripts\python.exe C:/Users/Daniil/PycharmProjects/stbp/LABS/kulyk10/main.py
RSA: LICENSE KEY   : jTmzxPUmCJ9m5Zvfanhk/CosrAWZMKdpaiYTXkHKgXuARN76cJN3qQ5FvhVssdmJ11hwPtbhK9PLKUcL+9/sPw==
RSA: IS VALID      : True
SHA1: LICENSE KEY   : oBYeButbocVKlXm53k40jTkX6kc=
SHA1: IS VALID      : True
```

**Рисунок 4 –** Результат роботи програми

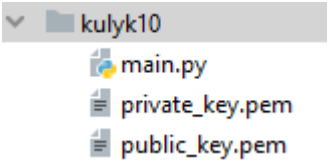


**Рисунок 5 –** Згенеровані ключі

**Висновки:** в результаті виконання лабораторної роботи було досліджено і порівняно існуючі механізми створення і перевірки валідності ліцензійних ключів.