Міністерство освіти і науки України Національний авіаційний університет Факультет кібербезпеки, комп'ютерної та програмної інженерії Кафедра комп'ютеризованих систем управління

Лабораторна робота № 2.2 з дисципліни «Захист інформації в комп'ютерних системах» на тему «Застосування криптографічних програмних бібліотек»

> Виконав: студент ФККПІ групи СП-425 Клокун В. Д. Перевірила: Супрун О. М.

1. МЕТА РОБОТИ

Ознайомлення з програмними бібліотеками криптографічних функцій.

2. ЗАВДАННЯ РОБОТИ

Використовуючи можливості криптографічних програмних бібліотек, створити програмний додаток, що виконує прості криптографічні операції.

3. ХІД РОБОТИ

Щоб виконати завдання лабораторної роботи, використаємо мову програмування Python і модуль стуртодтарну. Перш за все, встановимо необхідний модуль. Для цього створюємо віртуальне середовище за допомогою такої команди:

```
python -m venv venv
```

В результаті з'явиться віртуальне середовище під назвою «venv». Тепер активуємо його за допомогою такої команди для операційної системи Windows:

```
venv\Scripts\activate.bat
```

або її аналога для GNU/Linux:

```
source venv\bin\activate
```

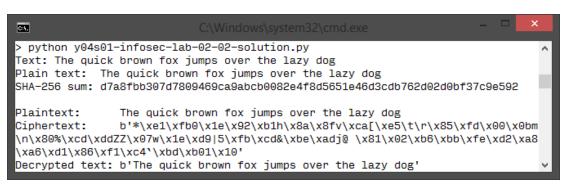
Тепер, знаходячись у віртуальному середовищі, встановлюємо необхідний модуль за допомогою такої команди:

```
pip install cryptography
```

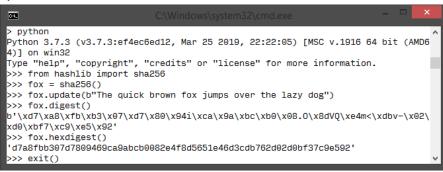
Завершивши виконання, команда встановить необхідний модуль. Коли модуль встановлений, розробляємо програмний додаток, який обчислить значення хеш-суми за допомогою алгоритму SHA-256, а також зашифрує і розшифрує повідомлення за допомогою автентифікованого шифрування із додатковими даними за алгоритмом ChaCha20Poly1305 (лістинг A.2, A.1).

Запускаємо розроблений додаток, вводимо текст і спостерігаємо результат (рис. 1).

Як бачимо, розроблений програмний додаток працює справно і дає правдоподібний результат. Порівнюємо хеш-суму, отриману в розробленому додатку, із хеш-сумою, отриманою від реалізації в стандартній бібліотеці hashlib мови програмування Python і бачимо, що значення співпадають, отже алгоритм обчислення хеш-суми реалізований правильно.



a)



б)

Рис. 1: Результати обчислення хеш-сум, шифрування і розшифрування: а — розробленого програмного додатку, б — модуля hashlib

4. Висновок

Виконуючи дану лабораторну роботу, ми знайомились з програмними бібліотеками криптографічних функцій.

А. Початковий код програмної реалізації модуля

Лістинг А.1: Файл залежностей розробленого програмного додатку

```
1 cffi==1.13.2
2 cryptography==2.8
3 pycparser==2.19
4 six==1.13.0
```

Лістинг А.2: Початковий код програмного модуля для шифрування повідомлення

```
import os
   from cryptography.hazmat.backends import default_backend
   from cryptography.hazmat.primitives import hashes
   from cryptography.hazmat.primitives.ciphers.aead import ChaCha20Poly1305
4
5
6
   def run compute hash(text):
7
        """Computes the SHA-256 sum of the given string.
8
9
        digest = hashes.Hash(hashes.SHA256(), backend=default_backend())
10
        digest.update(text.encode())
11
        res = digest.finalize()
12
        print(
13
            "Plain text: {}\n"
            "SHA-256 sum: {}\n"
15
            .format(
16
                text,
17
                res.hex(),
18
            )
19
        )
20
21
22
   def run_aead_chacha(plaintext, associated_data=""):
23
        """Encrypts and decrypts given plaintext using ChaCha20Poly1305 AEAD.
24
25
        data = plaintext.encode()
26
        aad = associated_data.encode()
27
        key = ChaCha20Poly1305.generate_key()
```

```
chacha = ChaCha20Poly1305(key)
29
        nonce = os.urandom(12)
30
        ciphertext = chacha.encrypt(nonce, data, aad)
31
        decrypted_text = chacha.decrypt(nonce, ciphertext, aad)
32
33
        print(
34
            "Plaintext:
                              {}\n"
35
            "Ciphertext:
                              {}\n"
36
            "Decrypted text: {}"
37
            .format(
38
                plaintext,
39
                ciphertext,
40
                decrypted_text,
41
42
        )
43
44
45
   if __name__ == "__main__":
46
        text = input("Text: ")
47
48
        run_compute_hash(text)
        run_aead_chacha(text)
49
```