Національний Технічиний Університет України

«Київський Політехнічний Інститут імені Ігоря Сікорського»

Інститут Прикладного Системного Аналізу

Кафедра Системного Проектування

Безпека інформаційних систем

Лабораторна робота №4

Роботу виконала:

Желєзнова В.С.

Група: ДА-81

Перевірив: Яременко В. С.

Київ – 2021

# Мета роботи

Ознайомитися з основними принципами хешування інформації та забезпечення надійності її передачі, а також підтвердження її достовірності за допомогою цифрового підпису.

# Завдання

1. Реалізувати на будь-якій об’єктно-орієнтовній мові програму хешування даних за алгоритмом SHA-1.
2. Реалізувати на будь-якій об’єктно-орієнтовній мові програму симетричного шифрування методом простих перестановок. Ширину таблиці методу простих перестановок визначити за формулою w = [10 + (v + g)%7], де v - номер студента в групі, g - номер групи.
3. Реалізувати програму асиметричного шифрування методом RSA.
4. Використовуючи програми з пунктів 1,2,3 імітувати передачу даних з перевіркою особи відправника з комбінованим шифруванням. Для цього:

4.1. Обрахувати SHA-1 хеш повідомлення;

4.2. Зашифрувати отриманий хеш симетричним шифруванням;

4.3. Зашифрувати ключ симетричного шифрування w асиметричним шифруванням за допомогою відкритого ключа, отриманого від “отримувача”;

4.4. Вважаючи особу власника відкритого ключа підтвердженою через “мережу довіри” стандарту OpenPGP, використати закритий ключ для розшифрування асиметричного шифру;

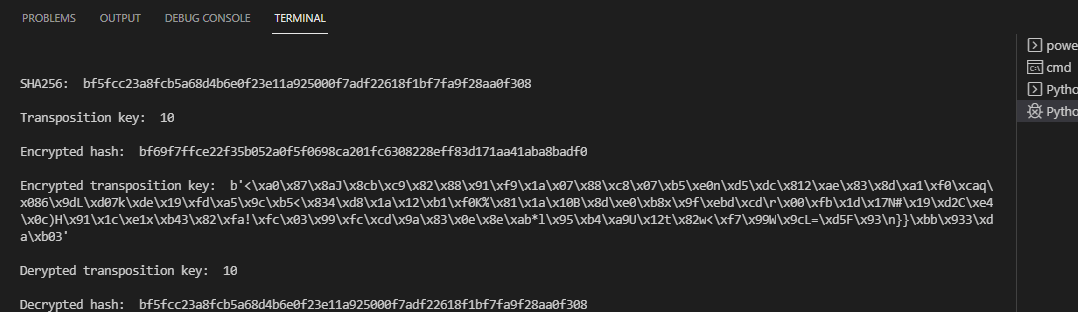
4.5. Використати результат в якості ключа для розшифрування симетричного шифру;

4.6. Вирахувати хеш “отриманого” повідомлення та порівняти з результатом розшифрування “симетричного шифру”.

1. Використати програму CrypTool для реалізації атаки “Днів народження”, самостійно обравши вихідний текст та текст підміни. Кількість символів хешу SHA-1, що має збігтись в тексті обрати за формулою l = [10 + (v + g)%7], де v - номер студента в групі, g - номер групи. Перевірити SHA-1 скорегованих текстів.
2. Оформити звіт по роботі.

# Хід роботи

1. Результати роботи програми:



Лістинг програми:

import math

from Cryptodome.PublicKey import RSA

from Cryptodome import Random

from Cryptodome.Cipher import PKCS1\_OAEP

from Cryptodome.Hash import SHA256

with open('10.txt', 'r') as file:

    data = file.read()

class RsaEncryption:

    @staticmethod

    def get\_keys():

        random\_generator = Random.new().read

        key\_pair = RSA.generate(1024, random\_generator)

        private\_key = key\_pair.export\_key('PEM').decode()

        public\_key = key\_pair.publickey().export\_key('PEM').decode()

        return private\_key, public\_key

    @staticmethod

    def encrypt(key, message):

        message = message.encode()

        key = RSA.importKey(key)

        cipher = PKCS1\_OAEP.new(key)

        encrypted = cipher.encrypt(message)

        return encrypted

    @staticmethod

    def decrypt(key, encoded\_message):

        key = RSA.importKey(key)

        decipher = PKCS1\_OAEP.new(key)

        decrypted = decipher.decrypt(encoded\_message)

        return decrypted

class TranspositionCypher:

    @staticmethod

    def encrypt(key, message):

        ciphertext = [''] \* key

        for col in range(key):

            pointer = col

            while pointer < len(message):

                ciphertext[col] += message[pointer]

                pointer += key

        return ''.join(ciphertext)

    @staticmethod

    def decrypt(key, encoded\_message):

        numOfColumns = math.ceil(len(encoded\_message) / key)

        numOfRows = key

        numOfShadedBoxes = (numOfColumns \* numOfRows) - len(encoded\_message)

        plaintext = [''] \* numOfColumns

        col = 0

        row = 0

        for symbol in encoded\_message:

            plaintext[col] += symbol

            col += 1

            if (col == numOfColumns) or (col == numOfColumns - 1

                                         and row >= numOfRows - numOfShadedBoxes):

                col = 0

                row += 1

        return ''.join(plaintext)

def main():

    transposition\_key = 10 + (10 + 81) % 7

    hash\_object = SHA256.new(data=data.encode())

    print('\n\nSHA256: ', hash\_object.hexdigest())

    encrypted\_hash = TranspositionCypher.encrypt(

        transposition\_key, hash\_object.hexdigest())

    transposition\_key = str(transposition\_key)

    print('\nTransposition key: ', transposition\_key)

    print('\nEncrypted hash: ', encrypted\_hash)

    private\_key, public\_key = RsaEncryption.get\_keys()

    encrypted\_tr\_key = RsaEncryption.encrypt(public\_key, transposition\_key)

    print('\nEncrypted transposition key: ', encrypted\_tr\_key)

    decrypted\_tr\_key = int(RsaEncryption.decrypt(

        private\_key, encrypted\_tr\_key).decode())

    print('\nDerypted transposition key: ', decrypted\_tr\_key)

    decrypted\_hash = TranspositionCypher.decrypt(

        decrypted\_tr\_key, encrypted\_hash)

    print('\nDecrypted hash: ', decrypted\_hash)

if \_\_name\_\_ == "\_\_main\_\_":

    main()

2. Реалізація атаки «Днів народження» через CrypTool

