МІНІСТЕРСТВО ОСВІТИ І НАУКИ, МОЛОДІ ТА СПОРТУ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ

Кафедра Кібербезпеки

ЛАБОРАТОРНА РОБОТА

з дисципліни «Методи реалізації криптографічних механізмів»

Тема: Порівняння бібліотек OpenSSL, crypto++, CryptoLib, PyCrypto для розробки гібрідної криптосистеми під Linux платформу.

Виконав:
студент 2 курсу
групи ФБ-11мн
Мохонько М.М.

Мета роботи:

Вибір базових бібліотек/сервісів для подальшої реалізації криптосистеми.

Завдання:

Підгрупа 2B. Порівняння бібліотек OpenSSL, crypto++, CryptoLib,

PyCrypto для розробки гібрідної криптосистеми під Linux платформу.

Хід роботи:

Обрані засоби профілювання:

Час:

- Python - memory_profiler
- C++ - valgrind

Пам’ять:

- Python/C++ - Внутрішні таймери

Використані бібліотеки та утиліти:

OpenSSL, Cryptopp, PyCryptodome(PyCrypto), memory-profiler,

valgrind, make, g++

Профілювання функцій бібліотеки Crypto++:

```
builder@Crypto:~/prof$ cat crypto++_size_prof
==2943== Memcheck, a memory error detector
==2943== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==2943== Using Valgrind-3.15.0 and LibVEX; rerun with -h for copyright info
==2943== Command: ./obj/crypto++_size
==2943==
==2943==
==2943== HEAP SUMMARY:
==2943==     in use at exit: 7,286 bytes in 10 blocks
==2943==   total heap usage: 12,628,367 allocs, 12,628,357 frees, 1,467,277,560 bytes allocated
==2943==
==2943== LEAK SUMMARY:
==2943==    definitely lost: 0 bytes in 0 blocks
==2943==    indirectly lost: 0 bytes in 0 blocks
==2943==      possibly lost: 0 bytes in 0 blocks
==2943==    still reachable: 7,286 bytes in 10 blocks
==2943==         suppressed: 0 bytes in 0 blocks
==2943== Rerun with --leak-check=full to see details of leaked memory
==2943==
==2943== For lists of detected and suppressed errors, rerun with: -s
==2943== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
```

```
builder@Crypto:~/prof$ cat crypto++_time_prof
pritate key: 30.2102
open key: 5.5043e-05
ciphertext: 0.00896258
plaintext: 0.833466
```

Профілювання функцій бібліотеки OpenSSL:

```
builder@Crypto:~/prof$ cat OpenSSL_size_prof
==3004== Memcheck, a memory error detector
==3004== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==3004== Using Valgrind-3.15.0 and LibVEX; rerun with -h for copyright info
==3004== Command: ./obj/OpenSSL_size
==3004==
==3004==
==3004== HEAP SUMMARY:
==3004==     in use at exit: 4,494 bytes in 34 blocks
==3004==   total heap usage: 1,343,871 allocs, 1,343,832 frees, 255,984,350 bytes allocated
==3004==
==3004== LEAK SUMMARY:
==3004==    definitely lost: 200 bytes in 2 blocks
==3004==    indirectly lost: 4,296 bytes in 37 blocks
==3004==      possibly lost: 0 bytes in 0 blocks
==3004==    still reachable: 0 bytes in 0 blocks
==3004==         suppressed: 0 bytes in 0 blocks
==3004== Rerun with --leak-check=full to see details of leaked memory
==3004==
==3004== For lists of detected and suppressed errors, rerun with: -s
==3004== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
```

```
builder@Crypto:~/prof$ cat OpenSSL_time_prof
private/open key: 63.8842
ciphertext: 0.0190879
plaintext: 0.528481
```

Профілювання функцій бібліотеки PyCrypto:

```
builder@Crypto:~/prof$ cat PyCrypto_size_prof
Filename: PyCrypto_size.py

Line #    Mem usage    Increment  Occurences   Line Contents
=============================================================
     6     40.8 MiB     40.8 MiB           1   @profile
     7                                         def memProf():
     8     40.8 MiB      0.0 MiB           1       msg = bytes(Settings.MESSAGE, 'utf-8')
     9    492.8 MiB      0.0 MiB        1001       for i in range(Settings.NUM_TESTS):
    10                                             # згенерувати відкритий/закритий ключ
    11    492.8 MiB    443.8 MiB        1000           key = RSA.generate(Settings.CRYPTO_RSA_KEY_LEN, e=17)
    12
    13                                             # зашифрувати повідомлення
    14    492.8 MiB      0.0 MiB        1000           encryptor = PKCS1_OAEP.new(key.publickey())
    15    492.8 MiB      3.1 MiB        1000           encrypted = encryptor.encrypt(msg)
    16
    17                                             # розшифрувати повідомлення
    18    492.8 MiB      0.0 MiB        1000           decryptor = PKCS1_OAEP.new(key)
    19    492.8 MiB      5.2 MiB        1000           decrypted = decryptor.decrypt(encrypted)
```

```
builder@Crypto:~/prof$ cat PyCrypto_time_prof
private/open key: 448.97037267684937
cipher text: 0.7617835998535156
open text: 2.21715688870544434
```

ВИСНОВКИ

У цій роботі були розглянуті бібліотеки, що реалізують криптографічні перетворення crypto++, OpenSSL, PyCrypto. З точки зору зручності та шкивдкості написання коду найкраще себе зарекомендовала PyCrypto, проте це в більшій мірі заслуга самої мови програмування. З точки зору профілювання найкраще себе показала бібліотека crypto++, яка відпрацювала в два рази швидше ніж OpenSSL, та на порядок краще ніж PyCrypto.

З точки зору реальних продуктів для бізнесу найкраще, на мій погляд, використовувати бібліотеку crypto++. В лабораторних умовах мені особисто більш зручно використовувати Python через більш інтуітивний процес реалізації певного криптографічного функціоналу. Саме тому у подальшому я буду використовувати PyCrypto.