Міністерство освіти і науки

Київський національний університет імені Тараса Григоровича Шевченка

Кафедра Мережевих та Інтернет технологій

**Звіт**

з практичної роботи **№7**

з дисципліни: “Основи інформаційної безпеки”

**«Асиметричне шифрування як засіб забезпечення конфіденційності інформації»**

Виконав:

Студент групи МІТ-21

Горбун О. М.

**Результати виконання завдань роботи:**

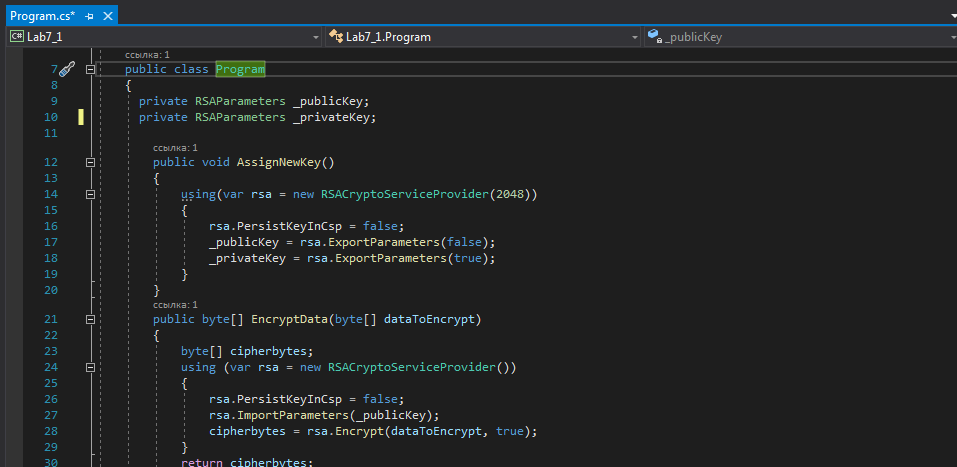
За завданнями практичної роботи було створено програми, які можна переглянути за посиланням:

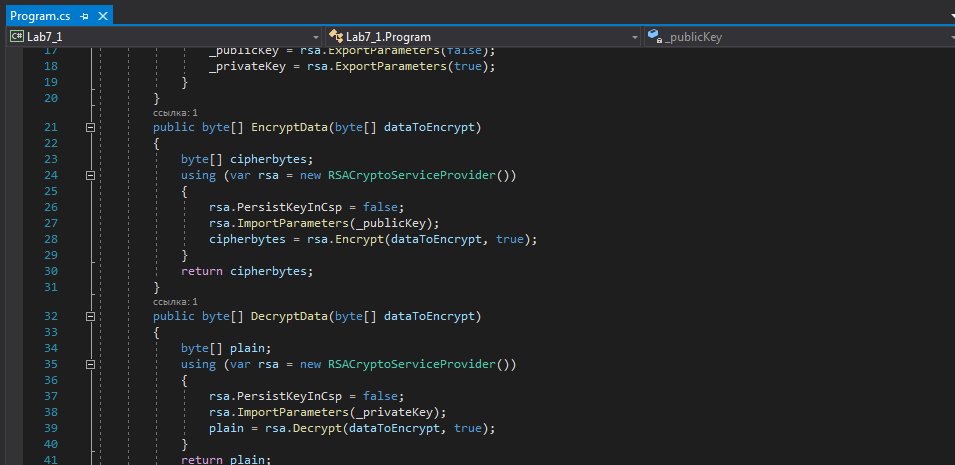
<https://github.com/OleksandrHorbun/Basics_Cybersecurity.git> (Посилання на відкритий репозиторій у веб-сервісі GitHub) завдання у вітці (Branch) Homework\_7.

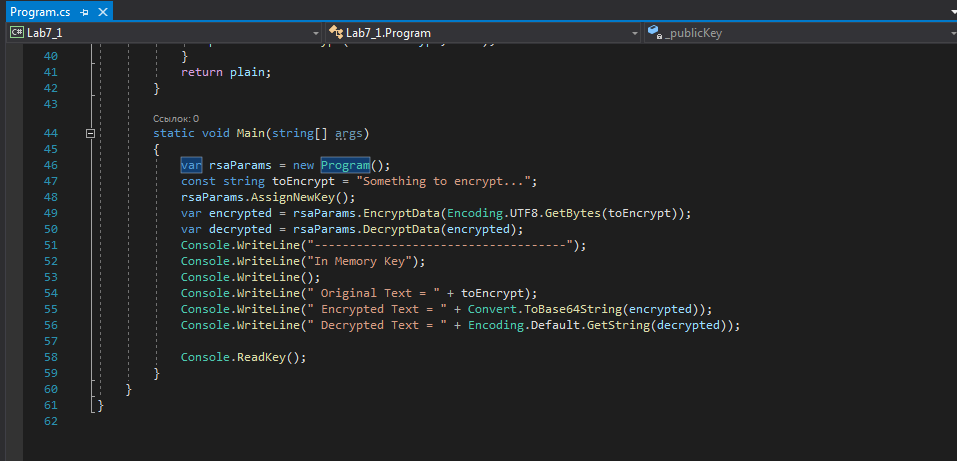
**I частина (перше завдання)**

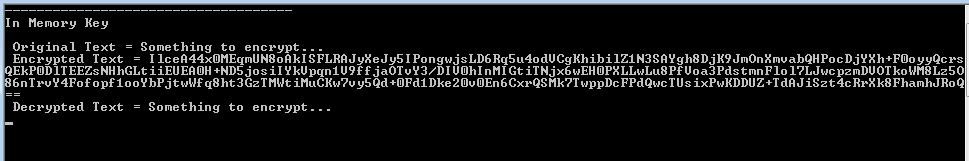
Як можна побачити змінна з параметрами є об’єктом класу *Program*. Саме в цьому відкритому (публічному) класі записані функції для шифрування та розшифрування, але окрім того в ньому прописано, що об’єкти цього класу будуть містити в собі обидва ключа для асиметричного шифрування: відкритий (публічний) та закритий (секретний або також приватний). Отже, в програмі вже наявний рядок, який буде зашифровано. Шифрування буде виконано за допомогою алгоритму RSA. Спочатку ключі генеруються за допомогою функції *AssignNewKey* (обидва ключі будуть розміром 2048 біт = 32 байт, ключі будуть збережені в пам’яті). Після цього починається шифрування: функція *EncryptData* приймає повідомлення та зашифровує його за допомогою відкритого ключа. Основний процес шифрування відбувається у функції *Encrypt* (функція з криптографічної бібліотеки). Результат повертається в основну програму. Одразу після цього, щоб побачити що при розшифруванні інформація не буде зіпсована, починається розшифрування за допомогою функції *DecryptData.* Розшифрування відбувається вже за секретним ключем, який функція бере з пам’яті. Очевидно що асиметричне шифрування вимагає виконання математичних операцій більшої складності, але саме наявність публічного та секретного ключів чудово забезпечує приватність повідомлення. Після дешифрування, отриманий початковий рядок повертається у основну частину програми та відбувається вивід отриманих результатів на екран.

Все описане вище можна побачити на знімках екрану:





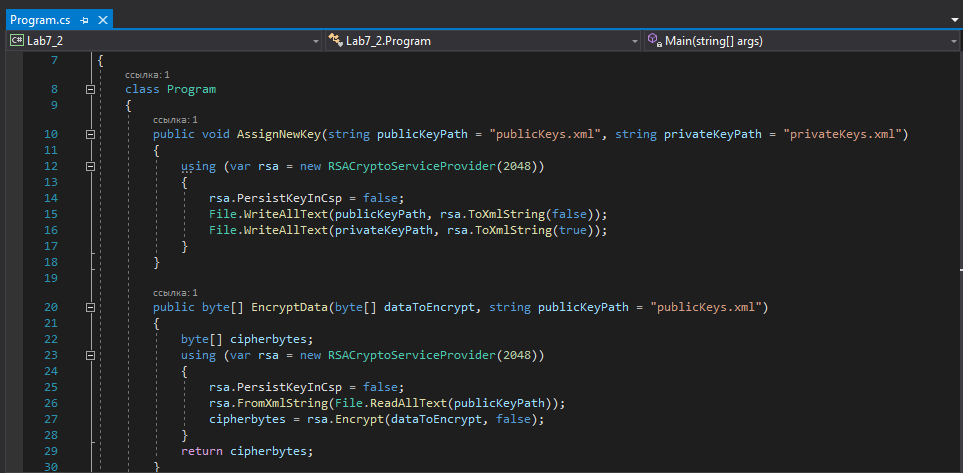


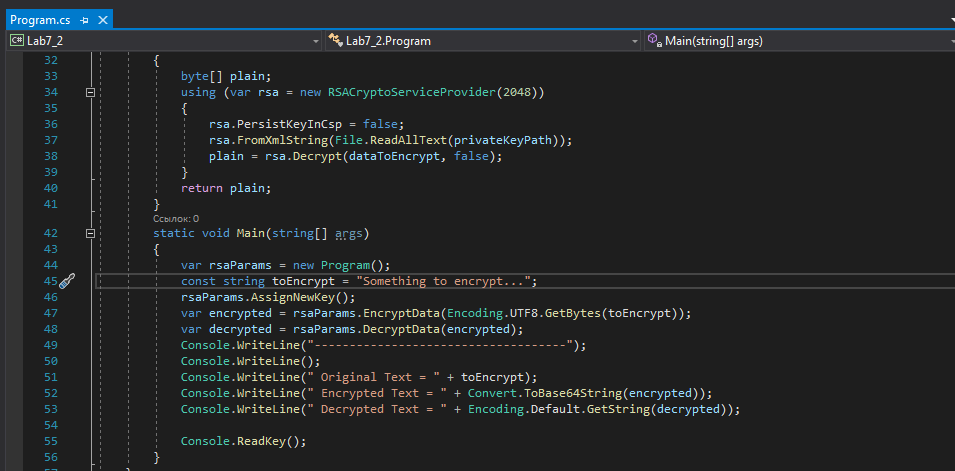


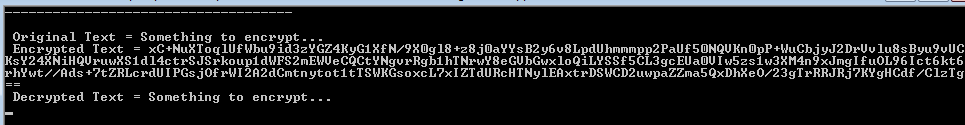
**IІ частина (друге завдання)**

Друга програма є копією першої за виключенням того, що тепер ключі зберігаються у файлах типу .xml . Разом з цим можна легко зашифровувати повідомлення за допомогою публічних ключів різних користувачів, просто «покласти» ключ у .xml файл. Більша частина наявного коду була пояснена вище (див. перше завдання).

Все описане вище можна побачити на знімках екрану:







**Висновок:**

Виконуючи цю роботу я навчився зашифровувати та розшифровувати дані за допомогою алгоритму асиметричного шифрування RSA, як зберігати ключі у памяті, у файлах та контейнерах ключів, у тому числі СSP (cryptographic service provider) Windows.