МІНІСТЕРСТВО ОСВІТИ І НАУКИ, МОЛОДІ ТА СПОРТУ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»
ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ
Кафедра Інформаційної Безпеки

Криптографія

Комп'ютерний практикум №4

Вивчення криптосистеми RSA та алгоритму електронного підпису; ознайомлення з методами генерації параметрів для асиметричних криптосистем

Мета:

Ознайомлення з тестами перевірки чисел на простоту і методами генерації ключів для асиметричної криптосистеми типу RSA; практичне ознайомлення з системою захисту інформації на основі криптосхеми RSA, організація з використанням цієї системи засекреченого зв'язку й електронного підпису, вивчення протоколу розсилання ключів.

Перевірив:

_____

Виконали:
студенти III курсу
групи ФБ-01
Приходько І.Ю.
та Сахній Н.Р.

Київ 2022

Постановка задачі:

1. Написати функцію пошуку випадкового простого числа з заданого інтервалу або заданої довжини, використовуючи датчик випадкових чисел та тести перевірки на простоту. В якості датчика випадкових чисел використовуйте вбудований генератор псевдовипадкових чисел вашої мови програмування. В якості тесту перевірки на простоту рекомендовано використовувати тест Міллера-Рабіна із попередніми пробними діленнями. Тести необхідно реалізовувати власноруч, використання готових реалізацій тестів не дозволяється.
2. За допомогою цієї функції згенерувати дві пари простих чисел p, q і p_1, q_1 довжини щонайменше 256 біт. При цьому пари чисел беруться так, щоб $pq \le p_1q_1$; p і q – прості числа для побудови ключів абонента A, p_1 і q_1 – абонента B.
3. Написати функцію генерації ключових пар для RSA. Після генерування функція повинна повертати та/або зберігати секретний ключ (d, p, q) та відкритий ключ (n, e) . За допомогою цієї функції побудувати схеми RSA для абонентів A і B – тобто, створити та зберегти для подальшого використання відкриті ключі (e, n) , (e_1, n_1) та секретні d і d_1.
4. Написати програму шифрування, розшифрування і створення повідомлення з цифровим підписом для абонентів A і B. Кожна з операцій (шифрування, розшифрування, створення цифрового підпису, перевірка цифрового підпису) повинна бути реалізована окремою процедурою, на вхід до якої повинні подаватись лише ті ключові дані, які необхідні для її виконання.

   За допомогою датчика випадкових чисел вибрати відкрите повідомлення M і знайти криптограму для абонентів A і B, перевірити правильність розшифрування. Скласти для A і B повідомлення з цифровим підписом і перевірити його.
5. За допомогою раніше написаних на попередніх етапах програм організувати роботу протоколу конфіденційного розсилання ключів з підтвердженням справжності по відкритому каналу за допомогою алгоритму RSA. Протоколи роботи кожного учасника (відправника та приймаючого) повинні бути реалізовані у вигляді окремих процедур, на вхід до яких повинні подаватись лише ті ключові дані, які необхідні для виконання. Перевірити роботу програм для випадково обраного ключа $0 < k < n$

Кожна з наведених операцій повинна бути реалізована у вигляді окремої процедури, інтерфейс якої повинен приймати лише ті дані, які необхідні для її роботи; наприклад, функція Encrypt(), яка шифрує повідомлення для абонента, повинна приймати на вхід повідомлення та відкритий ключ адресата (і тільки його), повертаючи в якості результату шифротекст. Відповідно, програмний код повинен містити сім високорівневих процедур: GenerateKeyPair(), Encrypt(), Decrypt(), Sign(), Verify(), SendKey(), ReceiveKey().

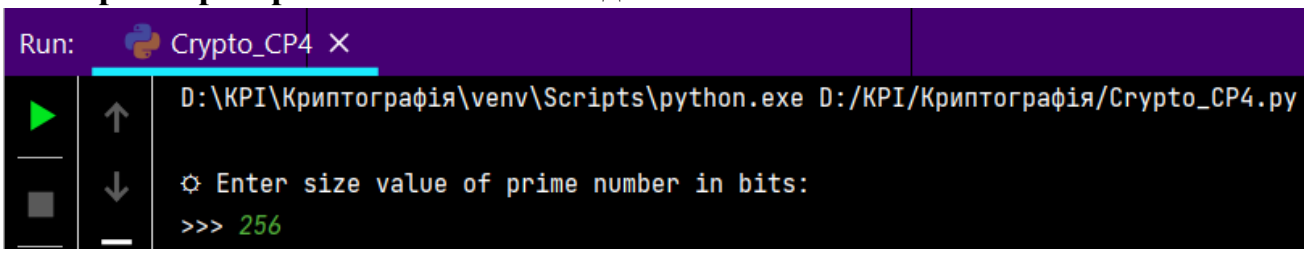
Кожну операцію рекомендується перевіряти шляхом взаємодії із тестовим середовищем, розташованим за адресою:

http://asymcryptwebservice.appspot.com/?section=rsa.

Наприклад, для перевірки коректності операції шифрування необхідно зашифрувати власною реалізацією повідомлення та розшифрувати його на сервері. А ось для того, щоб перевірити процес розшифрування, необхідно зашифрувати на сервері повідомлення та розшифрувати його локально за допомогою реалізованої власної програми.

Хід роботи:

1. Параметри криптосистеми RSA для абонентів A і B



1.1. *Дві пари великих випадково згенерованих простих чисел* p, q *та* p_1, q_1

```
[x]► Pair of prime numbers for user A
     p = 73658276694353937118748368055007068495544245084587683525780165251043332960347
     q = 90800893407433479359562734507428754067437046774732317962563954865575438337693

[x]► Pair of prime numbers for user B
    p1 = 10229192337822954537917013971530980086343115338785643969463495283968032622097 9
    q1 = 71704808744587266076116508744254837884781035588443982139709780823452442907139
```

1.2. *Пари чисел* n, e *та* n_1, e_1 *– відкриті ключі, а* d, d_1 *– секретні ключі*

```
[***]► Generation RSA key pair for user A
• Public key of user A:
     e = 52700238787444216947446505949868734818225736057416044231297726548640301618784190146281557097636162954102760882252477434972128269011610131487929555722684229
     n = 66882373306992735056582145066529102267833769352630701917561023189398221754104233176721853429779967805786151512746955981339930754046277602907673943644594 71
• Secret key of user A:
     d = 50797858591915617699494309187516331609483936533883654512170328908058038049680557144137617557234651449673135284956350453894021025529375450366799703335306 93
     p = 73658276694353937118748368055007068495544245084587683525780165251043332960347
     q = 90800893407433479359562734507428754067437046774732317962563954865575438337693

[***]► Generation RSA key pair for user B
• Public key of user B:
    e1 = 52919531321094323634955387029645257399918724853436697193707621944947390851481725944787784650633918539384988050865351758608038323883521072123622673799828 17
    n1 = 73348228019519245012042072622043466783677485594850094907200201431486896893191002232131595069337956476521424766018850084185974075802630846765606284906690 81
• Secret key of user B:
    d1 = 65242229362297624979286138005011565751007911476645206466684957606425869887841419535317317377737736783012228636616468057220537547317058761380482366923471849 3
    p1 = 10229192337822954537917013971530980086343115338785643969463495283968032622097 9
    q1 = 71704808744587266076116508744254837884781035588443982139709780823452442907139
```

1.3. *Чисельні значення прикладів ВТ, ШТ, цифрового підпису для* A *та* B

```
Message --> 18171965999233628979975509007400366641118552237605416833315664546985048367873087443066562529902146188836000612659275537791512765787225989699272445652700 2

• User B encrypt message for user A:  111267062022706605427979265999395040863237445765999436330181248656406555189418051523581632284085616639420399751110819030763673591735574662386146317467562 7
[]> Digital signature of user B:  69364190800159040012358231840184664521641702451052319271212067898836972870082598939426446886199405178031707755067927903645812221674566555134630085858376277
• User A encrypt message for user B:  19448120906730217862572025839515891353910423522127168173283988144582850172846619212820359927780372142145086715285061010999897389565123673698952826679114 21
[]> Digital signature of user A:  14961445132451173289698200878140696135867521291528666584098094100104109116748956238621255858099865858005032583050094817552388707477370264255184162010603 40

[*] User A decrypt message from user B after successful signature verification:  18171965999233628979975509007400366641118552237605416833315664546985048367873087443066562529902146188836000612659275537791512765787225989699272445652700 2
[*] User B decrypt message from user A after successful signature verification:  18171965999233628979975509007400366641118552237605416833315664546985048367873087443066562529902146188836000612659275537791512765787225989699272445652700 2
```

1.4. *Протокол розсилання ключів з підтвердженням справжності, чисельні значення характеристик на кожному кроці*

```
Key --> 31413346440459219840951040513248448317600063053906754820938849046926182510836833794619674672369977227851098037558795669243590833898252931883526952446668 7

> User A forms message (K1, S1) ⇔ (17301747315493451873114696166254606128211666078988274559626398748896665491425695757436412323005128842499310094512698489285842157209846403347293965366526 05, 15785096892232198179622166558429000585954592915269057606445 0
> User B using own secret key finds shared key ↓
[♦]► Key was successfully received from authenticated user: 31413346440459219840951040513248448317600063053906754820938849046926182510836833794619674672369977227851098037558795669243590833898252931883526952446668 7

> User B forms message (K1, S1) ⇔ (32577304123730273684866810435956211323317998851044941023486649027274132996974930106381358575188718921199925484996630403655937435572292736915424508333226, 6109225697094379311742201281628376738546390032966999816768150
> User A using own secret key finds shared key ↓
[♦]► Key was successfully received from authenticated user: 31413346440459219840951040513248448317600063053906754820938849046926182510836833794619674672369977227851098037558795669243590833898252931883526952446668 7


Process finished with exit code 0
```

Висновки:

У ході виконання лабораторної роботи були здобуті практичні навички захисту інформації на основі криптосхеми RSA. З використанням цієї системи було організовано засекречений зв'язок та процес верифікації повідомлення за цифровим підписом користувача.

Ми дізналися про алгоритми перевірки числа на простоту, використовуючи при цьому відповідні тести перевірки чисел на простоту. А також ознайомилися з методами генерації ключів для асиметричної криптосистеми типу RSA.

За протоколом розсилення ключів була перевірялася конфеденційсть надісланої інформації та автентичність відправника.