

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ, МОЛОДІ ТА СПОРТУ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»**

ФІЗИКО- ТЕХНІЧНИЙ ІНСТИТУТ

Кафедра інформаційної безпеки

КОМП'ЮТЕРНИЙ ПРАКТИКУМ №4

Варіант 2

з дисципліни

Криптографія

З теми: « Вивчення криптосистеми RSA та алгоритму електронного підпису;
ознайомлення з методами генерації параметрів для асиметричних криптосистем »

Перевірила:

Селюх П.В

Виконав студент групи ФБ-92

Андрієвич Дмитро Юрійович

Мета роботи: 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.

Завдання:

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 $1 < p, q$ довжини щонайменше 256 біт. При цьому пари чисел беруться так, щоб $pq \leq p_1q_1$; p і q – прості числа для побудови ключів абонента А, $1 < p < q$ – абонента В.
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 А і В – тобто, створити та зберегти для подальшого використання відкриті ключі (e, n) , (p, q) і n і секретні d і d_1 .
4. Написати програму шифрування, розшифрування і створення повідомлення з цифровим підписом для абонентів А і В. 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 А і В, перевірити правильність розшифрування. Скласти для А і В 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()`.

Код програми знаходиться у файлі **main.py**.

Хід роботи:

Спочатку я реалізував функцію пошуку простих чисел з заданого інтервалу, після цього за допомогою неї згенерував дві пари p та q . Далі я реалізував функцію генерації пар ключів зі знайдених раніше p та q . Далі мною було реалізовано функції шифрування та дешифрування, також було зроблено функцію, яка відповідає за створення цифрового підпису, та ще одну, за допомогою якої цей підпис можна підтвердити. Після цього були створені два абоненти, на яких і було перевірено функціонування програми.

Результати:

P , q , $p1$, $q1$:

```
01 p = (int) 119760247020606402673133015737918467650736056358490634604527260676878205406269
01 p1 = (int) 161212015152815478095697451959683108304629352140439752214094403329983296932549
01 q = (int) 194586324469886749603930283008533957225491142169316216710905542348194898374873
01 q1 = (int) 159831047856510498528933160854568684718302859853290668955593691611103927414243
```

Кандидати, що не пройшли перевірку:

```
204738898625899148716235980126034200831906246746206747357271314160231239393493 144715484670401819597711011327843329971423368388209431701698121414602935508037
219018609671090945905515859275898851091842138076522840678882086304485457679737 160299592399219964992183860169868407477467844900129154068773497706570565095023
```

```
177777544342858061728108151265441263288971763527254002869055724187905057945401 164552629160073115662644823563875377882428709992586718229230613755628780631521
124480851678834978031338248768887253244268447800714547217299893160572308844437 181047037987369393648342131012469236720013999191394038646100775079767536849039
```

Параметри абонента A:

```
A = (User) <__main__.User object at 0x000001AE9EA35D00>
01 decrypted_received_message = (NoneType) None
01 e = (int) 9136018193024886552533118285626753600471895811423464987980485589401997151812787112749630655627461406304790147006060865202804958853369655514182588231088977
01 e1 = (int) 1661790077428883170662995419018856994501076916442834844707219638961349180832979525379813224421817693192020831615724273477697982446701137906005209080178031
01 encrypted_msg = (int) 2140728040867909414608842733319223153362322487379268029115056965306027459075119243344540896296876432678804475064231494942712835171200038122044686481446722
01 msg = (int) 16845136786764907511157642689796241319672520164304237915881791802795313988143246736660175728959173010232717340265255352285598520418345227759044195261923293
01 n = (int) 23303706285345505351305245532465844864911276625259295175486377324068529884502269852408711637265672175299820342814470712084696776142959124380246510826278837
01 n1 = (int) 2576668530893414632923280193031582699859228522757963470570736944890998850509165591138617256006856966296743124229806630201735736953254466869018891152895407
> received_message = (list: 0) []
01 signature = (int) 2091545392107280165543563183069668960662644823408454570125914313898426867135010880728725635595720274835991411990650616333935874864736547818932507480924208
Protected Attributes
01 _d = (int) 21388129581762061491241229792832558850344762482528561813355533627940511261720232091648952783235117629841142242401120162191464393113587065728241876502174833
01 _p = (int) 119760247020606402673133015737918467650736056358490634604527260676878205406269
01 _q = (int) 194586324469886749603930283008533957225491142169316216710905542348194898374873
```

Параметри абонента В:

```
B = {User} <__main__User object at 0x00001AE9EA35EE0>
decrypted_received_message = (int) 16845136786764907511157642689796241319672520164304237915881791802795313988143246736660175728959173010232717340265255352285598520418345227759044195261923293
e = (int) 1661790077428883170662995419018856994501076916442834844707219638961349180832979525379813224421817693192020831615724273477697982446701137906005209080178031
e1 = (int) 9136018193024886552533118285626753600471895811423464987980485589401997151812787112749630655627461406304790147006060865202804958853369655514182588231088977
encrypted_msg = (int) 4010684729173348214071432205192463079645301655464447491416172765855756971075429256037078290322798860789591851181563626419059061466500278416321577275067151
msg = (int) 4181755895158065018632476570374178555491517333138107879717529389986365329366608331742010503078788875334944153247733678670506138491318043726148967035899667
n = (int) 25766685308934146329232801930315582699859228522757963470570736944890998850509165591138617256006856966296743124229806630201735736953254466869018891152895407
n1 = (int) 23303706285345505351305245532465844864911276625259295175486377324068529884502269852408711637265672175299820342814470712084696776142959124380246510826278837
received_message = (list: 2) [2140728040867909414608842733319223153362322487379268029115056965306027459075119243344540896296876432678804475064231494942712835171200038122044686481446722, 413267...
signature = (int) 8606511595253658884368991006782476536356812679070494146235116088565222740383997649180795096792532959704255437534913167567795719462960128645360023407013621
Protected Attributes
_d = (int) 237276750052180122516503832797152794773892477473053127375619034965117286009942823949449398907045511858397697442872456640510442478398209850515732825714465911
_p = (int) 161212015152815478095697451959683108304629352140439752214094403329983296932549
_q = (int) 159831047856510498528933160854568684718302859853290668955593691611103927414243
```

Вхідний текст А:

16845136786764907511157642689796241319672520164304237915881791802795313988143246736660175728959173010232717340265255352285598520418345227759044195261923293

Шифрований текст А:

2140728040867909414608842733319223153362322487379268029115056965306027459075119243344540896296876432678804475064231494942712835171200038122044686481446722

Цифровий підпис А:

20915453921072801655435631830696689606626448234084545701259143138984268671350108807287256355955720274835991411990650616333935874864736547818932507480924208

Вхідний текст В:

4181755895158065018632476570374178555491517333138107879717529389986365329366608331742010503078788875334944153247733678670506138491318043726148967035899667

Шифрований текст В:

4010684729173348214071432205192463079645301655464447491416172765855756971075429256037078290322798860789591851181563626419059061466500278416321577275067151

Цифровий підпис В:

8606511595253658884368991006782476536356812679070494146235116088565222740383997649180795096792532959704255437534913167567795719462960128645360023407013621

Перевірка роботи програми на прикладі вхідного тексту А (абонент А відправляє абоненту В текст, який було зашифровано відкритим ключем абонента В) за допомогою сайту <https://www.dcode.fr/rsa-cipher>:

Search for a tool

★ SEARCH A TOOL ON DCODE BY KEYWORDS:

★ BROWSE THE [FULL DCODE TOOLS' LIST](#)

Results

🚩 ✓ Décryptation using C,D,N

16845136786764907511157642689796241319672520
16430423791588179180279531398814324673666017
57289591730102327173402652553522855985204183
45227759044195261923293

Ad closed by Google

RSA CIPHER

Cryptography › Modern Cryptography › RSA Cipher

RSA DECODER

Indicate known numbers, leave remaining cells empty.

★ VALUE OF THE CIPHER MESSAGE (INTEGER) C=

★ PUBLIC KEY E (USUALLY E=65537) E=

★ PUBLIC KEY VALUE (INTEGER) N=

★ PRIVATE KEY VALUE (INTEGER) D=

★ FACTOR 1 (PRIME NUMBER) P=

★ FACTOR 2 (PRIME NUMBER) Q=

★ INTERMEDIATE VALUE PHI (INTEGER) Φ=

★ DISPLAY ☐ PLAINTEXT AS CHARACTER STRING
☐ COMPUTED VALUES (C,D,E,N,P,Q,...)
☒ PLAINTEXT AS INTEGER NUMBER
☐ PLAINTEXT AS HEXADECIMAL FORMAT

Результат дешифрування співпадає з вхідним текстом абонента А.

Висновки: під час виконання даної лабораторної роботи я ознайомився з тестами перевірки чисел на простоту і методами генерації ключів для асиметричної криптосистеми типу RSA. Також, я ознайомився з криптосистемою RSA та реалізував засекречений зв'язок з використанням цієї системи.