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

Криптографія

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

Перевірила:

Селюх П.В

Виконали студенти групи ФБ-92

Ханас Максим Любомирович

Гуманков Денис Максимович

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 1 p , q довжини щонайменше 256 біт. При цьому пари чисел беруться так, щоб pq p1q1 ; p і q – прості числа для побудови ключів абонента А, 1 p і q1 – 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) , (,) 1 n1 e та секретні d і d1.
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 М і знайти криптограму для абонентів А и 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().

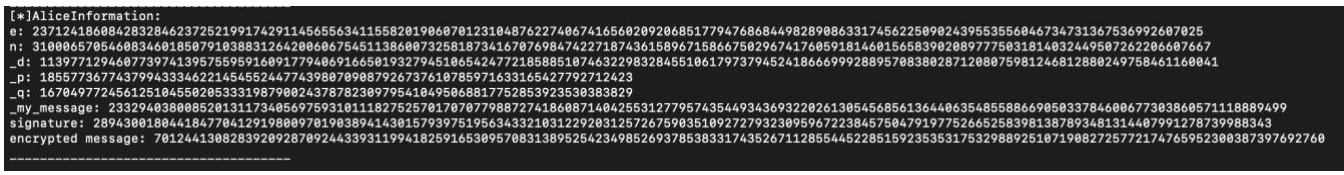
Код програми знаходиться у файлі **main.py**.

Хід роботи:

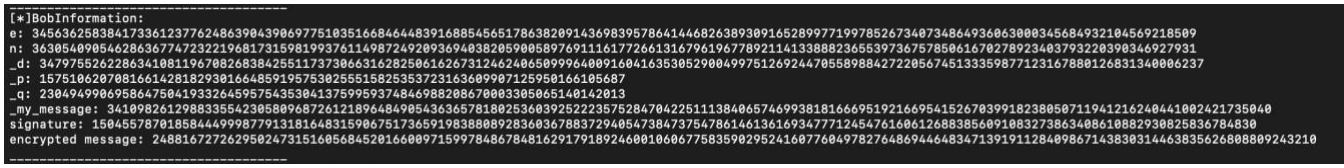
Спочатку ми реалізували функцію пошуку простих чисел з заданого інтервалу, після цього за допомогою неї згенерували дві пари p та q. Далі реалізували функцію генерації пар ключів зі знайдених раніше p та q. Далі нами було реалізовано функції шифрування та дешифрування, також було зроблено функцію, яка 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:

Параметри абонента A:



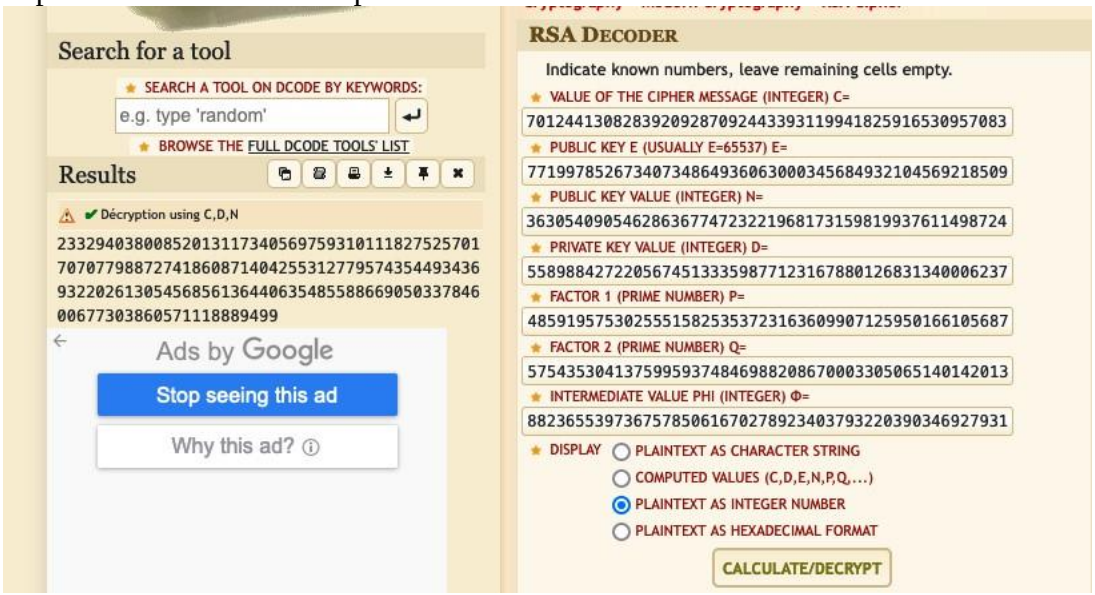
Параметри абонента B:



Приклад Комунікації:

```
1) Alice send message to Bob, and Bob verifies it
2) Bob send message to Alice, and Alice verifies it
3) Try yo intercept message using Dude as interface
4) Validate the message integrity(Teacher's request)
1
[*]Bob Verifying message :
Decrypted message: 233294038008520131173405697593101118275257017070779887274186087140425531277957435449343693220261305456856136440635485588669050337846006773038605711188894 99
Decrypted signature: 233294038008520131173405697593101118275257017070779887274186087140425531277957435449343693220261305456856136440635485588669050337846006773038605711188894 99
Bob receive: 233294038008520131173405697593101118275257017070779887274186087140425531277957435449343693220261305456856136440635485588669050337846006773038605711188894 99
[*]Bob Sent public public keys
[*]Bob Sent public public keys
[*]Alice Sent public public keys
[*]Alice Sent public public keys
```


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

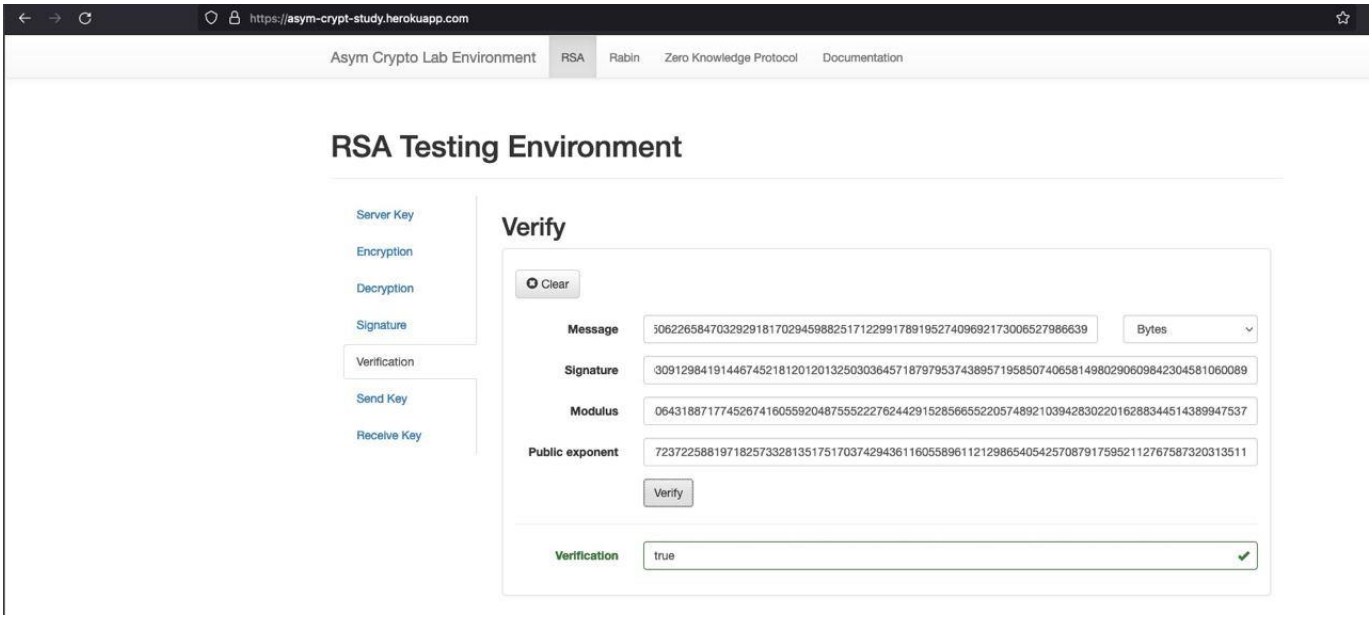


//Це приклад виконання, додаткової частини коду яка провіряє тільки цілісність без шифрування


```
[*]AliceInformation:
e: 331730407773847980640509554491089623094190604788460835206265203382034610389021662800510184947189241186272840985995607034884406984578586757670870941839746 67
n: 430556476847277491248425888576387089335893578311001108344370372334621430155911905862498987977359764961958628941996440641451160254071165452168597641452942709
_d: 190406678661786887401033451564795787722207154522032129298159448841517384094985082859838704543491454870483533474633671600949275844153112631304637692010786 3
_p: 220999656236852566659113689136076828495493963974326470713620847523400708277471 1
_q: 194822238268928998116434412502066855353968660450692621756635566233694832871219
_my_message: 955878909550887962065433807448443409392591827720701964418286100482976687017607020123260222616567903365595838921603715280676910312952293931999367666024766 3
signature: 209380520924540214269996938545982270952651656681344075906792820401972025941387325337309238120316508348298950326792123555361807378693485787842849314190894643
encrypted message: 47867774634549005872035007708940850425759352468951173452813970109665956754301157402524997154279621292522458053204102551189763206712187422230996453433012519

---------------------------------------
---------------------------------------
[*]BobInformation:
e: 194899914808843823095928698913715236353508328151215697123031864011642407415916092179466580619353132868373140155522584038851101347394848310949117516790442 9
n: 490198874059710247912851819347284959782709770862781950384833211557290210427855126913546106063799575731714216754409342393515785524760112212728242364859491 69
_d: 235630621593449400307727361162776921211586191664778800102989346046701832406716987114598645148540024220921751862175882263707593596697178755256141821671 8649
_p: 224703291676740373860900522982141048122983973069711817953900205310822945952571
_q: 218153846524381830861487060105838771155716416163512215457369804770716241535939
_my_message: 199450310187603047427245154793141606481248088047923448238296329617402291545816592240278804438746573679386302013845701931159503819221022840680520675512393 29
signature: 183557183122284417169442624155991307215011795715599690562672125820308955924462472763339765884725995902745928299172906395701292652026461559785487327823068 33
encrypted message: 193560500498164118823144498284571609145254195653674065636658130427483025485981701182526888886665196923567833554419124883751184575676171676715472326485498 00

---------------------------------------
1) Alice send message to Bob, and Bob verifies it
2) Bob send message to Alice, and Alice verifies it
3) Try yo intercept message using Dude as interface
4) Validate the message integrity(Teacher's request)
4
[*]Alice Sent public public keys
[*]Alice Sent public public keys
---------------------------------------
[*]AliceSending message without encryption !!!!
Message:
_my_message: 955878909550887962065433807448443409392591827720701964418286100482976687017607020123260222616567903365595838921603715280676910312952293931999367666024766 3
Dude receive: 955878909550887962065433807448443409392591827720701964418286100482976687017607020123260222616567903365595838921603715280676910312952293931999367666024766 3
```



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