

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ “КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ім. ІГОРЯ СІКОРСЬКОГО”

КРИПТОГРАФІЯ

КОМП’ЮТЕРНИЙ ПРАКТИКУМ №4

Вивчення криптосистеми RSA та алгоритму електронного

підпису; ознайомлення з методами генерації параметрів для

асиметричних криптосистем

Виконали:

Студенти групи ФБ-22

Орлов Антон, Ялбуган Федір

(бригада 7)

КИЇВ 2024

**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.

**Порядок виконання роботи**

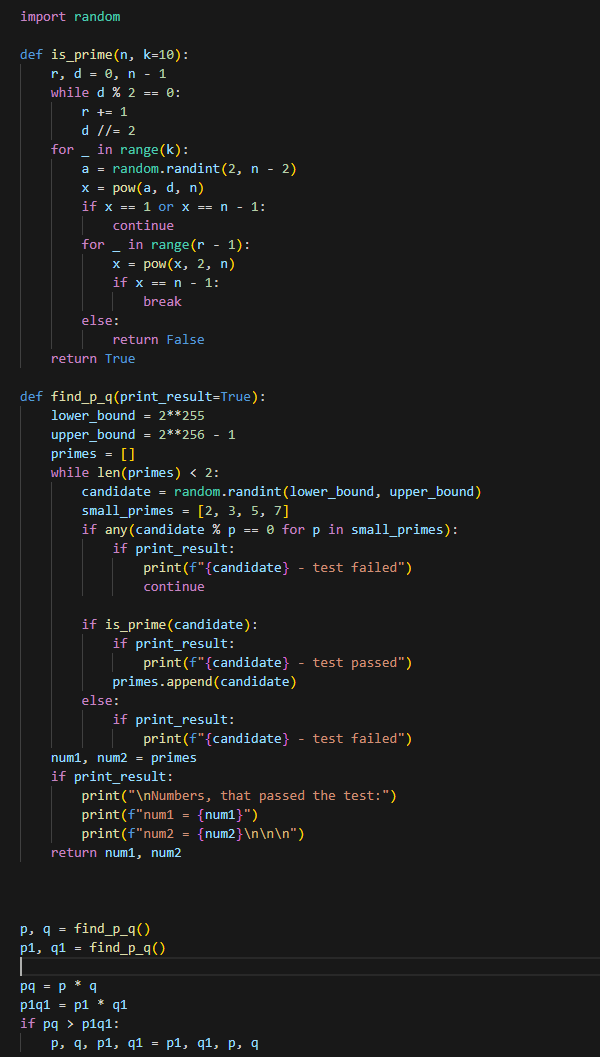
1. Написати функцію пошуку випадкового простого числа з заданого інтервалу або заданої довжини, використовуючи датчик випадкових чисел та тести перевірки на простоту. В якості датчика випадкових чисел використовуйте вбудований генератор псевдовипадкових чисел вашої мови програмування. В якості тесту 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 B.

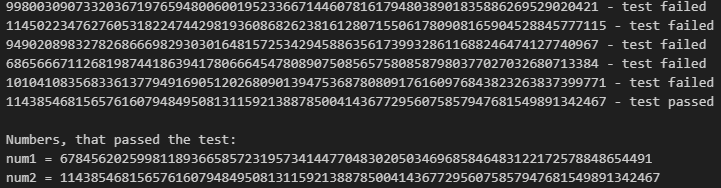
3. Написати функцію генерації ключових пар для RSA. Після генерування функція повинна повертати та/або зберігати секретний ключ (d, p,q) та відкритий ключ (n,e) . За допомогою цієї функції побудувати схеми RSA для абонентів А і B – тобто, створити та зберегти для подальшого використання відкриті ключі (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 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 А и B, перевірити правильність розшифрування. Скласти для А 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.

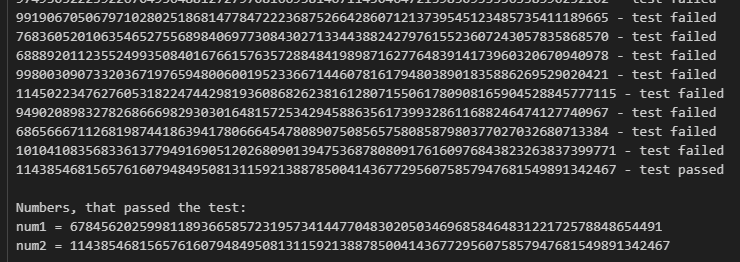
**Хід роботи**

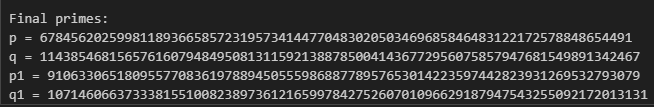
1. , 2. Функція пошуку випадкового простого числа з заданого інтервалу та генерація пар p, q:



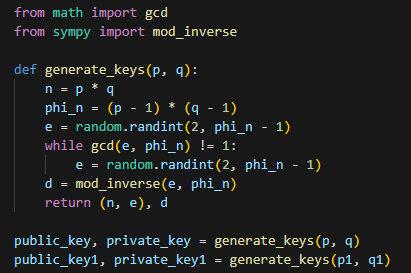
Випадково генеруються за допомогою random.randint числа з усіх можливих довжиною 256 біт, потім кожне перевіряється на простоту тестом Міллера-Рабіна, до тих пір, поки тест не пройдуть 2 числа – p, q. Далі p, q та p1, q1 міняються місцями між собою, якщо pq більше, ніж p1q1.

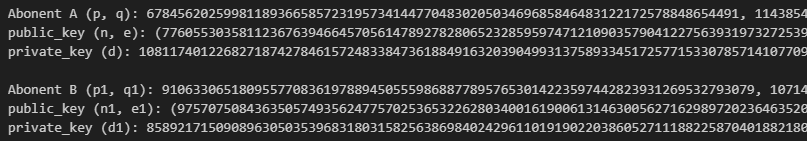


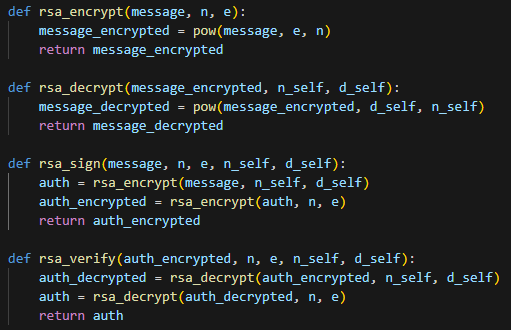




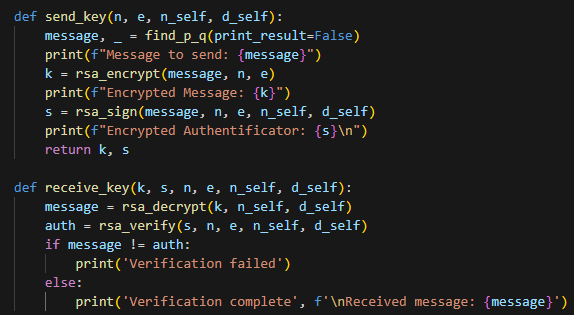
3. Функція генерації ключових пар для RSA:





4. Функції шифрування, розшифроування, створення цифрового підпису, перевірки цифрового підпису:  


5. Функції роботи протоколу конфіденційного розсилання ключів з підтвердженням справжності по відкритому каналу за допомогою алгоритму RSA.



**– опис кроків протоколу конфіденційного розсилання ключів з підтвердженням**

**справжності, чисельні значення характеристик на кожному кроці;**

1) Для абонентів A та B генеруються значення p, q, p1 , q1 відповідно:

Abonent A (p, q): 67845620259981189366585723195734144770483020503469685846483122172578848654491, 114385468156576160794849508131159213887850041436772956075857947681549891342467

Abonent B (p1, q1): 91063306518095577083619788945055598688778957653014223597442823931269532793079, 107146066373338155100823897361216599784275260701096629187947543255092172013131

2) Для абонентів A та B генеруються відкриті ключі (n, e), (n1, e1), а також секретні ключі d, d1:

public\_key (n, e): (7760553035811236763946645705614789278280652328595974712109035790412275639319732725395241279998067013415608548284739965428235452797732480427398272638569297, 288976268753249784722168377786300411425816860014700936549357176820728993513147548401024822637191612978317565358257499021103424642680778601983413649225497)

private\_key (d): 1081174012268271874278461572483384736188491632039049931375893345172577153307857141077093916564451286205744507542764930426365333471345641940971297802559313

public\_key (n1, e1): (9757075084363505749356247757025365322628034001619006131463005627162989720236463520094746751276999263651590229322933035343424362407271013550248912693920349, 5970176119551471963270139336348840766397873267870281694479170369250259896370246642860219899961384450905972793313180893450545551142040600778482153146671079)

private\_key (d1): 8589217150908963050353968318031582563869840242961101919022038605271118822587040188218001825979027658665744072033202680061070812065368870266023325313884219

3) Абонент А шифрує своє повідомлення відкритим ключем Абонента B (n1, e1):

On abonent A side

Message to send: 77633844615474466032948802040045145852619664267906824550427988761543202335487

Encrypted Message: 7047306672006644293351148232554494712090029461426221454944452084159717250357884265012630739104590003230293657913663769661073293451875152865992992161114416

4) Абонент А підписує повідомлення своїм секретним ключем (d), а потім, зашифровує відкритим ключем Абонента B (n1, e1), щоб зробити підпис:

Encrypted Authentificator: 4503363910727309112284084864530726244757947071152898054464263170247995013019894003158984196729195557158377142875835908051964547783489308813704135950811631

5) Абонент А передає зашифроване повідомлення (k) та цифровий підпис (s) Абоненту В

6) Абонент В розшифровує повідомлення своїм секретним ключем (d)

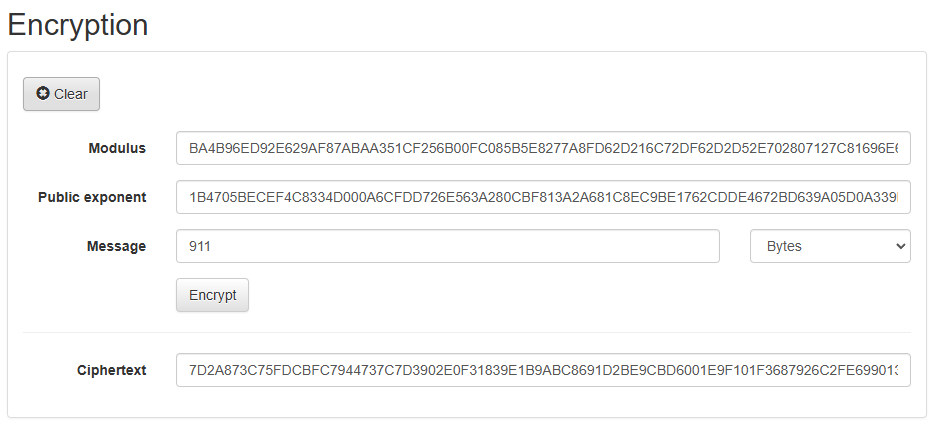
7) Абонент В розшифровує підпис спочатку своїм секретним ключем (d1) , а потім відкритим ключем Абонента А (n, e)

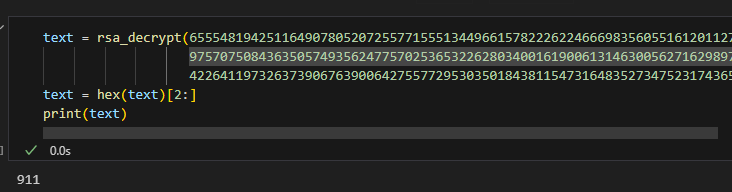
8) Абонент В верифікує підпис (порівнює підпис з розшифрованим повідомленням):

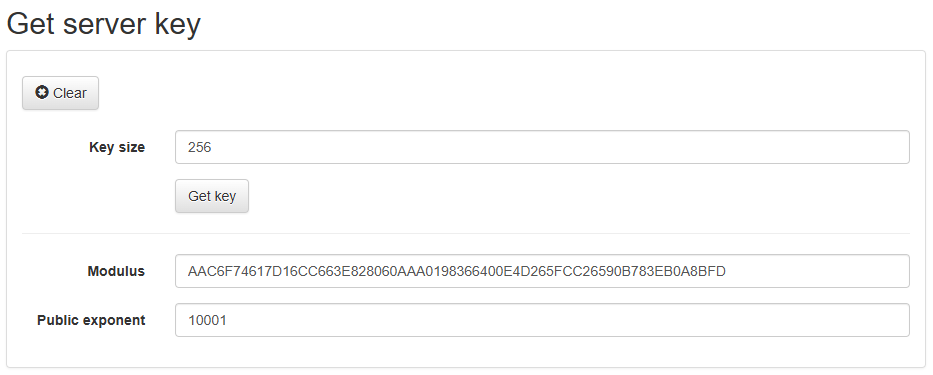
On abonent B side

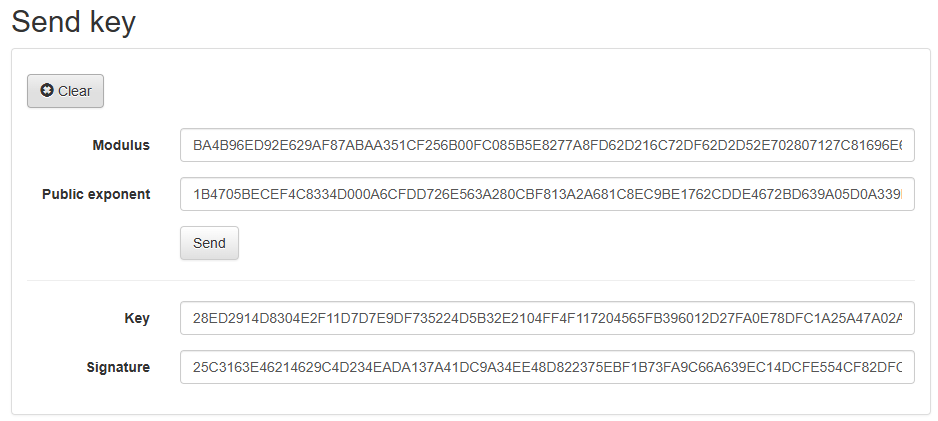
Verification complete

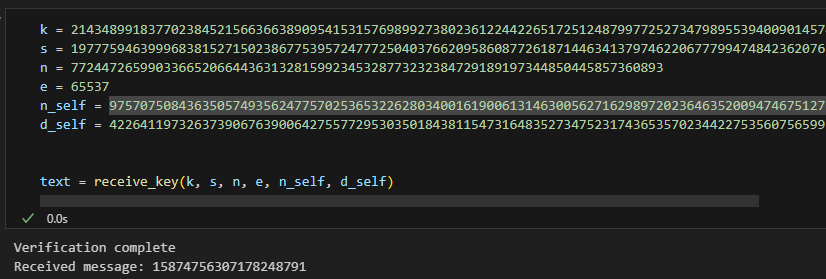
Received message: 77633844615474466032948802040045145852619664267906824550427988761543202335487

**- Перевірка шляхом взаємодії із тестовим середовищем:**  










**- Висновки**: ми ознайомились з тестами перевірки чисел на простоту і методами генерації ключів для асиметричної криптосистеми типу RSA. Практично ознайомились з системою захисту інформації на основі криптосхеми RSA, організація з використанням цієї системи

засекреченого зв'язку й електронного підпису, вивчення протоколу розсилання ключів.