

**Міністерство освіти і науки України Національний технічний університет
України "Київський політехнічний інститут імені Ігоря Сікорського"
Фізико-технічний інститут**

КРИПТОГРАФІЯ КОМП'ЮТЕРНИЙ ПРАКТИКУМ №4

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

**Виконали:
Вісловух Владислав
Ісаченко Федір
Варіант 9
Група: ФБ-06**

Київ - 2022

Мета та основні завдання 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.

Хід 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.

Для пошуку випадкового простого числа ми вибрали пошук з заданого інтервалу($\min=2^{**}256, \max=2*\min-2$)

Для генерування рандомних чисел використовували [randint](#) з [random](#).

В якості тесту на перевірку був вибран тест Міллера-Рабіна, також використовувалися попередня пробні ділення.

2. За допомогою цієї функції згенерувати дві пари простих чисел p, q і $1 \leq p, q$ довжини щонайменше 256 біт. При цьому пари чисел беруться так, щоб $p \nmid q$; $p \nmid q$ – прості числа для побудови ключів абонента А, $1 \leq p \nmid q$ – абонента В.

Згенерував пару де $p \nmid q$ для позбавлення подальших проблем.

$p =$
132405041945393801550531400419169696591181392528771732715668415064594306698
603

$q =$
152759408146090984034884488540376560975618711399302167507989714588627982065
373

p1 =

163514984698926303865492995117947175457118976699021058292165551453816147731421

q1 =

125717298368234364773928873183920154123633426595585713332882910114617404988461

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.

Приклад роботи коду:

```
e для абонента А : 8043264818144953645535087161923424643206399533763135266760509212725640758238497231617440085573610331602073510877478342892116910545014518652473353050147481
n для абонента А : 35706694380567503317153091132795547705487530106068704185126024752993059660681546455389079476278802728528658155337665723884560331202813990045246748185564089
d для абонента А : 2388526684451658409733601903016243888494998354538955009874459484339063955805369824999947341600893519257324967022602206157664181918959864737274342627126141
e для абонента В : 1461548842502325795649073945185294408989212218439372984316466748774923215428992172736123860788820706847725505492106172295909757956723282724911726510138521
n для абонента В : 374990840695713163521246834954375991202745458146555203991240472546040530588500407131600922977397966223370515315154437487572418825758118734493982451919259
d для абонента В : 5130587803328645133675595435620052134366519672452818821638020666913075772191549979370811746830553481851997665322150964572822512612417672639408622031622697
Message : 2581255178848069861233139982967522211355822206128061102471126578396730734869014414117308716923700455299781243609041812284743398106211159962614936725071405
Зашифроване повідомлення : 276792413083648515340407195145475725290437731286282480389485261224390626277976832821509413461930676139060858202573542646087069397443131989570100044448691163
Розшифроване повідомлення : 2581255178848069861233139982967522211355822206128061102471126578396730734869014414117308716923700455299781243609041812284743398106211159962614936725071405
Ключ k : 2574837925708587672109024222575450147004987576369082131912725801122080620280037282630326829609864030722943808665888080362707656408985635096554155838768214
Auth
Ключ k1 : 27028362524315876271724057588897208829161721899658910514348409276921847350521454834408986770093279752753957954068064626843475581076495287276559762598223682
Ключ s1 : 3474513632702782834971533134468075777738763204216397993198221233569748312189720648177128757195514807039180818126049751642569038829475869967663604258229439
['Verification ok', 2581255178848069861233139982967522211355822206128061102471126578396730734869014414117308716923700455299781243609041812284743398106211159962614936725071405]
```

Порівняння роботи сайту та мого коду.

Get server key

✖ Clear

Key size

256

Get key

Modulus

D9669737576D37A8B7CA4C1DF394514D51200E350C8E027D408EF3B79C4CBA9B

Public exponent

10001

Encryption

✖ Clear

Modulus

D9669737576D37A8B7CA4C1DF394514D51200E350C8E027D408EF3B79C4CBA9B

Public exponent

10001

Message

22

Bytes

Encrypt

Ciphertext

85322BCFE1025904FEF434D6BFB28FE34A45DE2B1837E9C49175882F7DF45820

```
# mod D9669737576D37A8B7CA4C1DF394514D51200E350C8E027D408EF3B79C4CBA9B
ACLE_n = 98333150198878732362049457536571285314575200836982636537534492909076445182619
# public exp (e) 10001
ACLE_e = 65537
# message 22
ACLE_message = 34

ACLE_encrypt = encrypt(ACLE_message,ACLE_e,ACLE_n)
# encrypt 60246253594311452391106401177082839448753614472186264481298290795565785831456
```

Decimal to Hexadecimal converter

From	To
Decimal	Hexadecimal

Enter decimal number

6024625359431145239110640117708283

10

= Convert

✕ Reset

↕ Swap

Hex number

85322BCFE1025904FEF434D6BFB28FE3
4A45DE2B1837E9C49175882F7DF45820

16

Як ми бачимо все шифрування працює коректно

Sign

Clear

Message

22

Bytes

Sign

Signature

35EADA246FE0A0C28D1EB49E92AE11A982E8890E2543ACB3F4B516C0253958F7

Verify

Clear

Message

22

Bytes

Signature

35EADA246FE0A0C28D1EB49E92AE11A982E8890E2543ACB3F4B516C0253958F7

Modulus

D9669737576D37A8B7CA4C1DF394514D51200E350C8E027D408EF3B79C4CBA9B

Public exponent

10001

Verify

Verification

true

✓

```
# signature 35EADA246FE0A0C28D1EB49E92AE11A982E8890E2543ACB3F4B516C0253958F7
ACLE_sign = 24387528751115011838962269802416685358045221117918323190434208470255678347511

ACLE_verify = check_sign(ACLE_message, ACLE_sign, ACLE_e, ACLE_n)
print(ACLE_verify)

/Users/qqakkashi/study/crypto/lab4/venv/bin/python /Users/qqakkashi/study/crypto/lab4/main.py
['Verification ok', 34]
```

Як ми бачимо написаний код працює відповідно, так отримує такі ж результати як на сайті.

Висновки

По ходу роботи ми ознайомилися та використали на практиці тест Міллера-Рабіна для перевірки текстів на простоту. Також дізналися про методи генерації ключів для криптосистеми RSA. Практично по-працювали з системою RSA організував секретний зв'язок та обмін даними за електронним підписом. Перевірили правильність нашої системи завдяки онлайн ресурсу.