



МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені Ігоря Сікорського»
«ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ»

КРИПТОГРАФІЯ

Комп'ютерний практикум №4

Виконали:
студенти 3-го курсу
групи ФБ-22
Власенко Г. В. та
Перебинос Р. О.
Бригада №2
Перевірів/-ла:

Київ – 2024

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

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

0. Уважно прочитати методичні вказівки до виконання комп'ютерного практикуму.
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 $pq \leq p_1q_1$; 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 и 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 $0 < k < n$.

Хід роботи

Згідно побудови криптосистеми RSA, на початку програми генеруємо ключі для абонента A : генеруємо два великі прості числа p і q довжини 256 біт (добуток яких, в парі з $e = 2^{16} + 1$, стає відкритим ключем) та знаходимо обернене (d , що стає приватним ключем) до e за модулем кількості взаємнопростих чисел з добутком p і q ($d = e^{-1} \bmod \phi(p \cdot q)$). Така ж процедура повторюється для абонента B , але ключі генеруються доки не стане виконуватися нерівність $pq \leq p_1q_1$. Процес генерації та отримані значення:

```
(kali@kali)-[~/Desktop/crypro-24-25/lab4/perebynos_fb-22_vlasenko_fb-22_cp4]
$ python3 main.py
Modulus length: 512

A: p=78134344167276603766376043384082604287417724910364956744478443487859918693987, q=88297650976499744544089137040891266800172116779248665776843426980794772377839
Invalid B: p=78509763456494282719037983895808984226433589721601306619874247956592865080519, q=69490547279290904975740398350207704491805042837136264730730853793056548
209203
Invalid B: p=10688098509868203329447908916504320750966888970884016992224761475997454117627, q=6105687393652744038007088969835397437361441003826972605860425231958770
1567863
Invalid B: p=69059337939451261299839780643609166730327731524938817385350391193992190937539, q=72984376749906202002984428154511785257245362884563297930126580298018480
705287
Invalid B: p=67694005226892738077422693627391504477681463608873321959501745068446082093739, q=10050013760682290040935951257012862432355713718127367634195715207681436
1367499
B: p=100191126057317725028789801870979451372557016073573806821536444881254509809907, q=85162664643686457041319735010041768640530288184970134256014937036593699052099

A public key: (65537, 68990790505598981320176982933977068338875371674440069051562422308992302902879872015573528558515645431939926809865041874443840546407510118790212
55381354093)
A private key: (44341823362022068315127254453895998788014532335486283909698526824532306679815003197506655911444843107570913981280128465480674312917874025747326341803
26681, 78134344167276603766376043384082604287417724910364956744478443487859918693987, 88297650976499744544089137040891266800172116779248665776843426980794772377839)

B public key: (65537, 85325432686926651166863123494927842489023982826300466870268912123567625589222961407980072555167342654382297228430096318657312705971680971037626
82579344793)
B private key: (66223326267917727567671586684918514857320366914258712595270278089709130347677367774101138665812852631326136641599969366788121250799097549192309306232
29733, 100191126057317725028789801870979451372557016073573806821536444881254509809907, 85162664643686457041319735010041768640530288184970134256014937036593699052099)
```

У криптосистемі використовувалося значення $e = 2^{16} + 1 = 65537$.

Згенеровані числа та параметри криптосистеми для абонента A :

p	78134344167276603766376043384082604287417724910364956744478443487859918693987
q	88297650976499744544089137040891266800172116779248665776843426980794772377839
n	6899079050559898132017698293397706833887537167444006905156242230899230290287987201557352855851564543193992680986504187444384054640751011879021255381354093
d	4434182336202206831512725445389599878801453233548628390969852682453230667981500319750665591144484310757091398128012846548067431291787402574732634180326681

Згенеровані числа та параметри криптосистеми для абонента B :

p	100191126057317725028789801870979451372557016073573806821536444881254509809907
q	85162664643686457041319735010041768640530288184970134256014937036593699052099
n	8532543268692665116686312349492784248902398282630046687026891212356762558922296140798007255516734265438229722843009631865731270597168097103762682579344793

<i>d</i>	662233262679177275676715866849185148573203669142587125952702780897091303 476773677741011386658128526313261366415999693667881212507990975491923093 0623229733
----------	--

Тепер, побудувавши криптосистему, можемо зашифрувати та розшифрувати повідомлення (зашифруємо з публічним ключем отримувача, розшифруємо – з його приватним ключем):

```
Test RSA encryption and decryption:
Message for A: 3394620972512114221566943770160809713111880921987643799461866623511112387346121590345613829439926109865954477193917281568334231870855649874903605948948701
Encrypted message for A: 366233742818643289946566811236291956145075909335990083076886245359387433847922221587082249864938762731206526283586520458102908716001381203045427131163852
Decrypted message for A: 3394620972512114221566943770160809713111880921987643799461866623511112387346121590345613829439926109865954477193917281568334231870855649874903605948948701

Message for B: 4829448863538812773016655942300420163864893007043860345157482300710638402926484118441547235518908129207590208334114655108436986463371124496213727039595949
Encrypted message for B: 1562268050066208254338694926896658202643360442952843954283941948915289267935381432688186781973677616054810029378925108819584590554878936701051193992980045
Decrypted message for B: 4829448863538812773016655942300420163864893007043860345157482300710638402926484118441547235518908129207590208334114655108436986463371124496213727039595949
```


Для абонента *A*:

<i>msg</i>	339462097251211422156694377016080971311188092198764379946186662351111 238734612159034561382943992610986595447719391728156833423187085564987 4903605948948701
<i>enc</i>	366233742818643289946566811236291956145075909335990083076886245359387 433847922221587082249864938762731206526283586520458102908716001381203 045427131163852
<i>dec</i>	339462097251211422156694377016080971311188092198764379946186662351111 238734612159034561382943992610986595447719391728156833423187085564987 4903605948948701


Для абонента *B*:

<i>msg</i>	482944886353881277301665594230042016386489300704386034515748230071063 840292648411844154723551890812920759020833411465510843698646337112449 6213727039595949
<i>enc</i>	156226805006620825433869492689665820264336044295284395428394194891528 926793538143268818678197367761605481002937892510881958459055487893670 1051193992980045
<i>dec</i>	482944886353881277301665594230042016386489300704386034515748230071063 840292648411844154723551890812920759020833411465510843698646337112449 6213727039595949

Перевірка на ресурсі <https://www.dcode.fr/rsa-cipher>:
Для абонента A:










Search for a tool

★ SEARCH A TOOL ON dCODE BY KEYWORDS:
e.g. type 'sudoku' 

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

Results






  Decryption using C,D,N

339462097251211422156694377016080971311188092
198764379946186662351111238734612159034561382
943992610986595447719391728156833423187085564
9874903605948948701

RSA Cipher - [dCode](#)

Tag(s) : Modern Cryptography, Arithmetics

Share

dCode and more

dCode is free and its tools are a valuable help in games, maths, geocaching, puzzles and problems to solve every day!


RSA CIPHER

Cryptography > Modern Cryptography > RSA Cipher

RSA DECODER

Indicate known numbers, leave remaining cells empty.


★ VALUE OF THE CIPHER MESSAGE (INTEGER) C=

3662337428186432899465668112362919561450759093359... 


★ PUBLIC KEY E (USUALLY E=65537) E=

65537 

★ PUBLIC KEY VALUE (INTEGER) N=

6899079050559898132017698293397706833887537167444... 

★ PRIVATE KEY VALUE (INTEGER) D=

4434182336202206831512725445389599878801453233548... 

★ 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

 **CALCULATE/DECRYPT**

Для абонента B:



Search for a tool

★ SEARCH A TOOL ON dCode BY KEYWORDS:

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

Results

🔍 ✓ Decryption using C,D,N

482944886353881277301665594230042016386489300
704386034515748230071063840292648411844154723
551890812920759020833411465510843698646337112
4496213727039595949

RSA Cipher - [dCode](#)

Tag(s) : Modern Cryptography, Arithmetics

Share



dCode and more

dCode is free and its tools are a valuable help in games, maths, geocaching, puzzles and problems to

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

 CALCULATE/DECRYPT

Далі – складаємо повідомлення із цифровим підписом та перевіряємо:

```
Test RSA signature and verification:
Message for A: 3248229287353330410683019275729704030977093059576665688153678329094447810921640374858977480568006716498932738599046630274891345777524300861820974273161349
Signature by A: 237291477662699558496657242883600754332963336591850109258594544475611471105364794329195228022663786786552638967624340475212637923956758294252424293230453
Decrypted signature by A: 3248229287353330410683019275729704030977093059576665688153678329094447810921640374858977480568006716498932738599046630274891345777524300861820974273161349
Verified by A: True

Message for B: 1966472729094273258493729042378362772821589773112539864725381657959292812444175975473835093386765524900546556354933741775272171261506572984541422176087451
Signature by B: 165451620138454951879819103308978699137552379452937017233753255028417683014810264803507373438866851626061246691194550663602135649554185568192522496835732
Decrypted signature by B: 1966472729094273258493729042378362772821589773112539864725381657959292812444175975473835093386765524900546556354933741775272171261506572984541422176087451
Verified by B: True
```

I, нарешті, протокол конфіденційного надсилання ключів:

```
Test RSA key sharing algorithm:
Key: 2705844863815073148360330133439330297480848737501760470037488703094715150675745895360889605804858066471596277118422804004843798810557050383001506536473866

Sending:
Encrypted key: 112140804461551274095111095132019067556985459131821805026255792769279555840093820259001049341489570894013219344273079579593593095504626587167845027149
468
Key signature: 424838806617823679397671312696530670604760767757171293937801845804003860505694788795092198443319644145494976687183261184687337749338938970305591454921
6118
Encrypted key signature: 422499118048472600644342727875538954694395616573055011551569397056498016419217581338572932777684367380164397747107004316209838940379759289
1437573843883

Receiving:
Decrypted key: 270584486381507314836033013343933029748084873750176047003748870309471515067574589536088960580485806647159627711842280400484379881055705038300150653647
3866
Decrypted key signature: 42483880661782367939767131269653067060476076775717129393780184580400386050569478879509219844331964414549497668718326118468733774933893897030
55914549216118
Decrypted key signature by A: 270584486381507314836033013343933029748084873750176047003748870309471515067574589536088960580485806647159627711842280400484379881055705
0383001506536473866
Verified: True
```

Опис кроків протоколу:

1. Генерується секретне (у нашому випадку випадкове) значення k .

k	270584486381507314836033013343933029748084873750176047003748870309471513866
-----	---

2. A зашифровує згенерований ключ із публічним ключем B .

k_I	11214080446155127409511109513201906755698545913182180502625579276927955468
-------	--

3. A підписує згенероване значення ключа k зі своїм приватним ключем.

S	424838806617823679397671312696530670604760767757171293937801845804003866118
-----	---

4. A зашифровує підписане значення ключа S із публічним ключем B та “відправляє” повідомлення (k_I, S_I) B .

S_I	42249911804847260064434272787553895469439561657305501155156931437573843883
-------	--

5. B отримує та дешифрує зашифроване значення ключа k_I за допомогою власного приватного ключа.

k	270584486381507314836033013343933029748084873750176047003748870309471513866
-----	---

6. B дешифрує зашифроване підписане значення ключа також за допомогою власного приватного ключа.

S	424838806617823679397671312696530670604760767757171293937801855914549216118
-----	---

7. B перевіряє підпис A (або зашифровує підписане значення ключа з публічним ключем A , те саме).

k	270584486381507314836033013343933029748084873750176047000383001506 536473866
-----	---

Висновки

У ході лабораторної роботи ми навчилися будувати криптосистему RSA, за допомогою якої шифрували та дешифрували повідомлення, підписували повідомлення та підтверджували їх справжність, а також побудували протокол конфіденційного розсилання ключів.