Бочок О.С., Павелко В.С. *ФБ-03*

КОМП'ЮТЕРНИЙ ПРАКТИКУМ №4

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 $p, q \leq p_1, q_1$; p і q – прості числа для побудови ключів абонента А, p_1 і q_1 – 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), (e_1, n_1) та 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 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 $0 < k < n$.

Хід роботи

Task 1-2

```
task 1-2
Composite:
990601193876158993246362344489366196264536264424254023822802235518359424946 96
Composite:
139230614231855521034761286016247287468116268379506180614540417775226987943 1
Composite:
605999034965115941195150212417954704935220551440846955204002979416271026771 84
Composite:
171339415736025575942248169329141599173713794997841362414121059205544352142 5
Composite:
394789784516829694010751074753274658744293453059066889389039111423582072684 19
Composite:
571380411167929068521479228459616841942016875336348332659111245650946250454 81
Composite:
```

```
p,q:
[201617642908466014109887278074511222032140505802283027541398024068521441998 99,
 113044314136873406122816720128598521898971684432007237799770476012432393496 267]
p1,q1:
[669511794336636384487328048698872959296925262279504227220827204670678334785 47,
 115684333144685551722961209027538431872029654816273276917096018093055968747 343]
```

p = 201617642908466014109887278074511222032140505802283027541398024068521441998 99

q = 113044314136873406122816720128598521898971684432007237799770476012432393496267

p_1 = 669511794336636384487328048698872959296925262279504227220827204670678334785 47

q_1=115684333144685551722961209027538431872029654816273276917096018093055968747343

За допомогою функції **findDigitIntervOrLen()** генеруємо числа заданої довжини, або в заданому проміжку. Перевіряємо їх на простоту за допомогою алгоритму Мілера-Рабіна **millerRabin().**

Потім формуємо пари чисел, та перевіряємо умову $pq \leq p_1 q_1$.

Task 3

```
A secret key:
(129475122248160215950148451242070502858674823006455167525443400168068256310116736197581494224731261846421244947182992130858700625701079012263644593726867
 1, 201617642908466014109887278074511222032140505802283027541398024068521441998 99,
 113044314136873406122816720128598521898971684432007237799770476012432393496267)
A open key:
(138494487869005263615868485488532497840733928539556693537940582842682890328628822041146842652295414296486768499027003935628584311788723650668586704402907
 9,
 227917281604805988767020351758014789572859770875640678779788015666212289621314914701943809673769118897528619399138561357463522078989145255110797395827703
 3)
B secret key:
(737751731410652263503646864581516235659271104449190287817996691905837420549969971342091462353249008986454053314643661414846310761285650814706639954429863
 9, 669511794336636384487328048698872959296925262279504227220827204670678334785 47,
 115684333144685551722961209027538431872029654816273276917096018093055968747343)
B open key:
(713925900211270442950119099486214613154803914724549533269054648338506129562887426627721873146573658866918425632886386757565278607378838060713781566287899
 1,
 774520254603356409515740633339288458499595371365733078957295291005595795011791213794722972285161792863670989355996785863680092656041805387487419385375062
 1)
```

Секретний ключ складається з (d, p, q), а відкритий = (n, e)

Генерація ключів реалізована функцією **rsa()**:

$$n = pq$$

$$\varphi(n) = (p-1)(q-1)$$

e вибирається в межах:

$$2 \le e \le \varphi(n) - 1 \quad \text{таке, що} \quad \gcd(e, \varphi(n)) = 1$$

d – обернений по модулю до e:

$$ed \equiv 1 (\mathrm{mod}\ \varphi(n)).$$

A secret key:

d =
12947512224816021595014845124207050285867482306045516752544340016806825631011673
61975814942247312618464212449471829921308587006257010790122636445937268671

p =

20161764290846601410988727807451122203214050580228302754139802406852144199899

q =
11304431413687340612281672012859852189897168443200723779977047601243239349626 7

A open key:

n =
13849448786900526361586848548853249784073392853955669353794058284268289032862882
20411468426522954142964867684990270039356285843117887236506685867044029079

e =
22791728160480598876702035175801478957285977087564067877978801566621228962131491
47019438096737691188975286193991385613574635220789891452551107973958277033

B secret key:

d =
73775173141065226350364686458151623565927110444919028781799669190583742054996997
13420914623532490089864540533146436614148463107612856508147066399544298639

p =
66951179433663638448732804869887295929692526227950422722082720467067833478547

q =
11568433314468555172296120902753843187202965481627327691709601809305596874734 3

B open key:

n =
71392590021127044295011909948621461315480391472454953326905464833850612956288742
66277218731465736588691842563288638675756527860737883806071378156628789971

e =
77452025460335640951574063333928845849959537136573307895729529100559579501179121
37947229722851617928636709893559967858636800926560418053874874193853750621

Task 4

```
task 4
Message
1348079985381789652642983110367289674417818283327542327661144774513997805978869936798381898827500018805499933535855249163516985264395096838466883922000886
Cryptograma A
724163577551123015053975795185508957842798326703470684544403654892228953953472244426733657678848539061461170079209443161098671515664779774123345795472 96
Cryptograma B
591082385294832137495067849528100233742198842612379868126206233881786206912209293273914693007127929278453976716565149657061915067927483441671381193159692 2
Decrypt A
1348079985381789652642983110367289674417818283327542327661144774513997805978869936798381898827500018805499933535855249163516985264395096838466883922000886
Decrypt B
134807998538178965264298311036728967441781828332754232766114477451399780597886993679838189882750001880549993353585524916351698526439509683846688392200088 6
Digital sign A
138613479904008428089997743711366869903444130226921275088092942006849799688586055844833514899716170301140134901379383747461468486038813272378541374454412 1
Digital sign B
1141857188951785493689734875789076042302642930078286417032638282745347735379475416891556308448108245443923950861227756045071463314017627269166852760724947
Digital sign verification A
True
Digital sign verification B
True
```

Генерується відкрите повідомлення M, в межах:

$$0 \le M \le n-1$$

Криптограма обчислюється за допомогою формули $C = M^e \bmod n$, що реалізована в функції: **rsaEncr().**

(e, n) – беруться з відкритого ключа.

Message

134807998538178965264298311036728967441781828332754232766114477451399780597886993679838189882750001880549993353585524916351698526439509683846688392200088 6

Cryptograma A

724163577551123015053975795185508957842798326703470684544403654892228953953472244426733657678848539061461170079209443161098671515664779774123345795472 96

Cryptograma B

591082385294832137495067849528100233742198842612379868126206233881786206912209293273914693007127929278453976716565149657061915067927483441671381193159692 2

Розшифрування криптограми обчислюється за формулою $M = C^d \bmod n$, яка реалізована у функції rsaDecr().

(d, n = p*q) – особистий ключ.

Decrypt A

134807998538178965264298311036728967441781828332754232766114477451399780597886993679838189882750001880549993353585524916351698526439509683846688392200088 6

Decrypt B

134807998538178965264298311036728967441781828332754232766114477451399780597886993679838189882750001880549993353585524916351698526439509683846688392200088 6

Цифровий підпис обчислюється за формулою $S = m^d \bmod n$, та повертає пару (m,S), що реалізовано у функції rsaDigitalSign().

Digital sign A

13861347990400842808999774371136686990344413022692127508809294200684979968858605584483351489971617030114013490137938374746146848603881327237854137445441 21

Digital sign B

11418571889517854936897348757890760423026429300782864170326382827453477353794754168915563084481082454439239508612277560450714633140176272691668527607249 47

Task 5

```
A k:
187249608027446236685718660278462311802924393989097412304899890129246324686449330548674665511089854855674157394781535296322250270999572267777398194004032 4
A S:
226581791956341733479939438196973770346776612370178804648271961093779279411061002954541847630702464867460726951761416353022368556680322364486136964114510
message (k1, S1):
(505860377842177170084862143566348885739769202274307093554096375758999856430770636374311083336870748770247737400435930299009618063006886687058328622089278
 7,
 157143105856834929184717281741443304322286883387440832684800177168216576378543424306371128206589923475396738589933715185150773445731924871203725369283517
 4)
B k:
187249608027446236685718660278462311802924393989097412304899890129246324686449330548674665511089854855674157394781535296322250270999572267777398194004032 4
B S:
226581791956341733479939438196973770346776612370178804648271961093779279411061002954541847630702464867460726951761416353022368556680322364486136964114510
Verify A k:
187249608027446236685718660278462311802924393989097412304899890129246324686449330548674665511089854855674157394781535296322250270999572267777398194004032 4
Verified
```

Абонент А має свій відкритий та закритий ключі, відкритий ключ Абонента В.

Секретне значення k генерується в межах $0 < k < n$.

k:

187249608027446236685718660278462311802924393989097412304899890129246324686449330548674665511089854855674157394781535296322250270999572267777398194004032 4

Значення S обчислюється за формулою $S = k^d \bmod n$, (d, n = p*q) – особистий ключ А.

S:

226581791956341733479939438196973770346776612370178804648271961093779279411061002954541847630702464867460726951761416353022368556680322364486136964114510

Повідомлення складається з (k_1, S_1), які обчислюються по формулам:

$k_1 = k^{e_1} \bmod n_1, \quad S_1 = S^{e_1} \bmod n_1$, ($e_1$, n_1) –відкритий ключ В.

Повідомлення:

k_1 = 505860377842177170084862143566348885739769202274307093554096375758999856430770636374311083336870748770247737400435930299009618063006886687058328622089278 7

S_1 =
157143105856834929184717281741443304322286883387440832684800177168216576378543424306371128206589923475396738589933715185150773445731924871203725369283517 4

Абонент В має свій відкритий та закритий ключі, відкритий ключ Абонента А.

Абоненту В приходить повідомлення від А

В за допомогою свого секретного ключа знаходить значення k та S, за формулами:

$k = k_1^{d_1} \bmod n_1, \quad S = S_1^{d_1} \bmod n_1$, ($d_1$, $n_1 = p_1 * q_1$) – особистий ключ В.

k =

187249608027446236685718660278462311802924393989097412304899890129246324686449330548674665511089854855674157394781535296322250270999572267777398194004032 4

S =

226581791956341733479939438196973770346776612370178804648271961093779279411061002954541847630702464867460726951761416353022368556680322364486136964114510

За допомогою відкритого ключа А, Абонент В перевіряє підпис А:

$k = S^e \bmod n$

Verify A k:

187249608027446236685718660278462311802924393989097412304899890129246324686449330548674665511089854855674157394781535296322250270999572267777398194004032 4

Verified

Висновок

Під час виконання комп’ютерного практикуму №4 ознайомились з тестами перевірки чисел на простоту і методами генерації ключів для асиметричної криптосистеми типу RSA. Окрім цього, практично дослідили систему захисту інформації на основі криптосхеми RSA, а також організували з використанням цієї системи засекреченого зв'язку й електронного підпису. І, нарешті, вивчили протокол розсилання ключів.

(код програми та результати експериментів прикріплюються 😈)