МІНІСТЕРСТВО ОСВІТИ І НАУКИ, МОЛОДІ ТА СПОРТУ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»
ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ

Лабораторна робота №4

з дисципліни **Криптографія** на тему:

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

Перевірив: Чорний О.М. _____	Виконали: Студенти групи ФБ-91 Красний П.В. Прищепа М.О.

Київ 2021

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 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().

Кожну операцію рекомендується перевіряти шляхом взаємодії із тестовим середовищем, розташованим за адресою http://asymcryptwebservice.appspot.com/?section=rsa.
Наприклад, для перевірки коректності операції шифрування необхідно а) зашифрувати власною реалізацією повідомлення для серверу та розшифрувати його на сервері, б) зашифрувати на сервері повідомлення для вашої реалізації та розшифрувати його локально

Хід роботи:

Кандидати на ключ, які не підійшли

key_candidates.txt

```
1  899983907326634193436096776891637565843823611642280302113164301404973555029343
2  554186947869851237090340616441357977381668157827845050756467812040222621685073
3  919761399614834235128092525651437417761263753694161162091007548151649913537090
4  192845084823780850924158067492616312803762390218302729539735260135807527330630
5  650652895403736228979157138143408161176311543209194964122819338167479969441720
6  955024493769370748831096922162666081479583162087337547355553671591844516495 42
7  793626555632473183122963294930488097584271388729050811985957006420689005216481
8  962046200717061474926560060296400184670706221276679009675851141161365575677200
9  667166370154961588381014381775014784144250038581367945011893403991244840642117
10 974818114466095864104648781409596789834678388858073155211671795135028113762447
11 185412940891749150166856898891849858061186630328447095965838280612767686113124
12 850418393365304732398101425954990457730755139311062105825279269216012022777433
13 663989805773614106449643134282546661732014837143874758172406516273356893931912
14 699108746768372327363684734815887037604396929224967566780758309740138956072941
15 160131726389398522397118876323781540603722611485611967280043722755310778747777
16 966969042153926493639674251158335473225426756323085728434988369860373580669604
17 879957796484558733105289026211421766311028901197832115319994652016188054937208
18 313833379621359178596023932035126484922218173847390651335606118863385517863211
19 228628715254760902952655271854400805158204452728010489411114437445532329191284
20 471645611842639282453923192561435942736678646342216702351521066697997196504397
21 657205865348897173905800637186854246440257144127115356359319292467802316130275
22 738440688620679968689403144321036105771412622321398461944131724585837057462320
23 742646284140285027600529282824010599407089929482601399701689509547416530693609
24 599059558068556241054177776953966335280492286108944199542404596384099454530326
25 875906590504304070728275451867386959483788151110695006233610346366914239429761
26 869443603388442810119115263990463216092058354703670218049314952253429707194672
27 211163669172504195165111014352920197696992925026566841141842091522929491954705
28 864252914957254945761429554134763692835959410399650925402052852371310329068219
29 978327409072569189970934416701110072835687281571066025765562530345518583461983
30 227939071501180972517029244709131001209665461001380185083733431758271593501458
31 726494680917617598049250575690193251832640171619976149858578349042271658066760
32 765068692795977360766503160278673244009510752506601075936822428418563052451228
33 687515276387634113244724669727433315929652031194862017865932236734970037875470
34 908077545675475757170683379922156674848727023580297497357668802273668431945165
```

Тут наведені перші 34 такі кандидати для прикладу, всі неправильні кандидати знаходяться у додатковому файлі Key_Candidates.txt

Ті кандидати, що пройшли відбір та параметри криптосистеми

p1, q1:

27331923179633773604819917661130622405820492301951094024943326402919
6519071319
26117761386498990932978239193681804036346690715993541900093084825309
8320090649

p2, q2:

93477750915117903101964942841701202127703993929749757958658835442346
6351284761
31587844941230090277138462301195793554070088347360466753370882926036
4332510809

n, e, d for abonent A

71384864783979569572492668757634084565049142596488405940626834128853
81474086286728646866779972384690992458772815507157184853196962244502
6293434341375996031
49924776786702368031551084112848300450397035666576361949378986945151
44267268634777929499743670760436846057008152817980460701300885090732
8012898717677337027
64516489901482087386553546785374773175173975647324564873754576880380
58775630867378590904299554477104251773129033301370189474985634557955
6349156246717782987

n1, e1, d1 for abonent B

29527607013616734975322711532162901760338917286093077429344574774566
92186977514547726730209916636838658810259923065911075668062341701112
66299562978069481649
19961816231488149535863037205618162318507776847790401909304069854518
49042381896830310266958912611514405517269036045108149392419989799347
55152091606952466251
15694246813535965300143800835491015407800592526781894619937456054493
00571082075119874558759692285445428236650162832371545521369213420418
47184733115480323171

Робота програми

For abonent A:

Message:
29263297733666571655920296070602410223291257017745027833083441871145
71268622170569572250095556623198360091745343554743125622331507102754
9191057235544050636
Encrypted message:
21488776175699821212385176534826825858768629293822110734863032598650
86862124618365368787532764201881967871580425426304419043573076227456
787173051825837114
Decrypted message:
29263297733666571655920296070602410223291257017745027833083441871145
71268622170569572250095556623198360091745343554743125622331507102754
9191057235544050636
Decryption is successful
Verification state: OK

For abonent B:

Message:
14862203492881970125029992931465226791733618506081582049763041416459
43870124988853348691208485682855927870832519586512339559209180260256
98398906434776032181
Encrypted message:
14730221954745442147699519897017384065048832944482896125523496302913
98134127588784745425282918823200921371100778714179472599241845822120
73323274203152015846
Decrypted message:
14862203492881970125029992931465226791733618506081582049763041416459
43870124988853348691208485682855927870832519586512339559209180260256
98398906434776032181
Decryption is successful
Verification state: OK

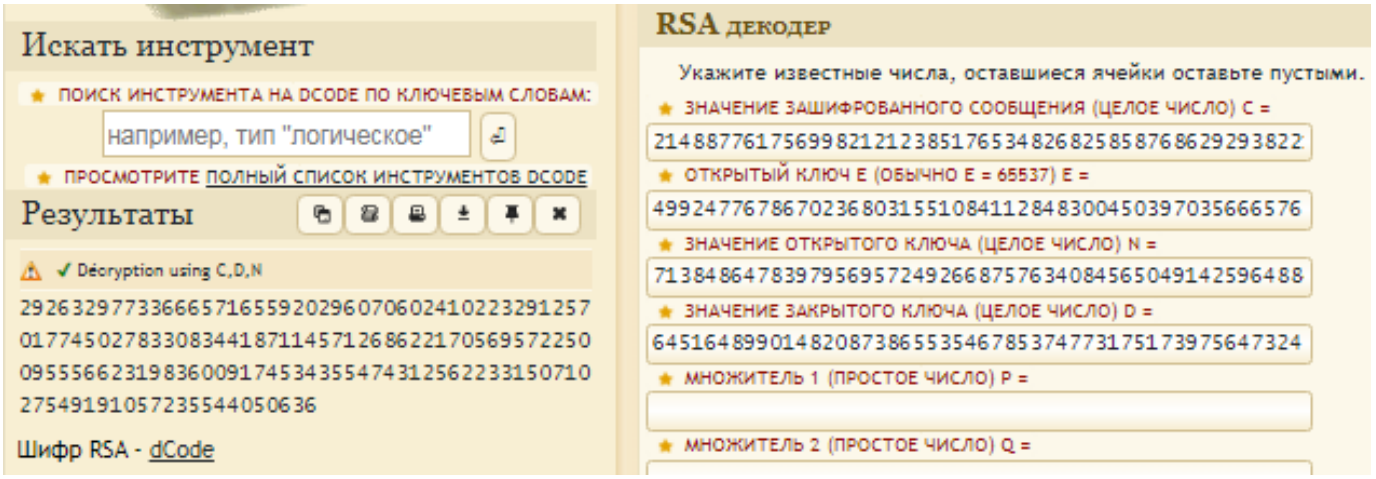
A generated a secret value k:
47563032811035282533753172351644064119962787604896931826380327314319
77891196812956340147903568019824258286861993555896549496307607304266
6468660931008911075
A created a message (k1, s1):
(44928680486677040710523189061967012635890769609279753878449577522659
61738607829091372390302152318741060500867012883845316699227730065181
1329062079127112695,
12471122949955597552097961078590937798654641885599873145777695696364

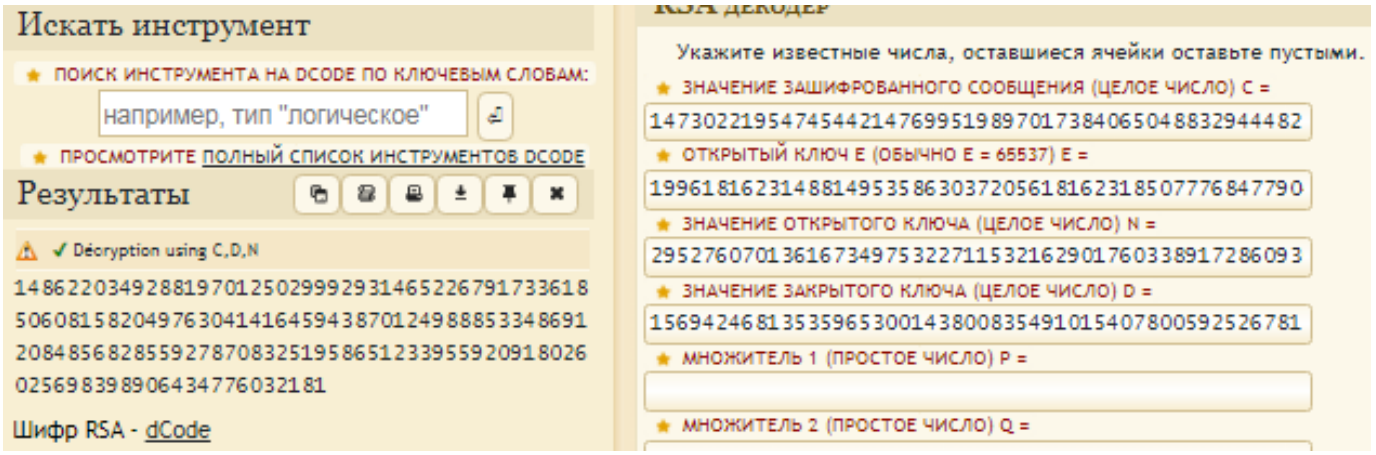
22444124999976268464317669022458634774753737107341246270428914002512
49859821314902563082)

B received (k, s):

(47563032811035282533753172351644064119962787604896931826380327314319
77891196812956340147903568019824258286861993555896549496307607304266
6468660931008911075,
36002817021021235249994892580989354688660035368485793378786522615906
96034727999150246711404384530199899133512953923611066439271647310725
0058335654448686936)

Перевірка





Обмін з сайтом

A :

N =

82be046e9cf4d9abced3a5337f36dd31a72dbfc0c9a9cce5103788a50ed1e55c7f77aca8
ddaf8e8ca3ab8c55ddd4f461dd0b9cb337290b777032bef64f74d663

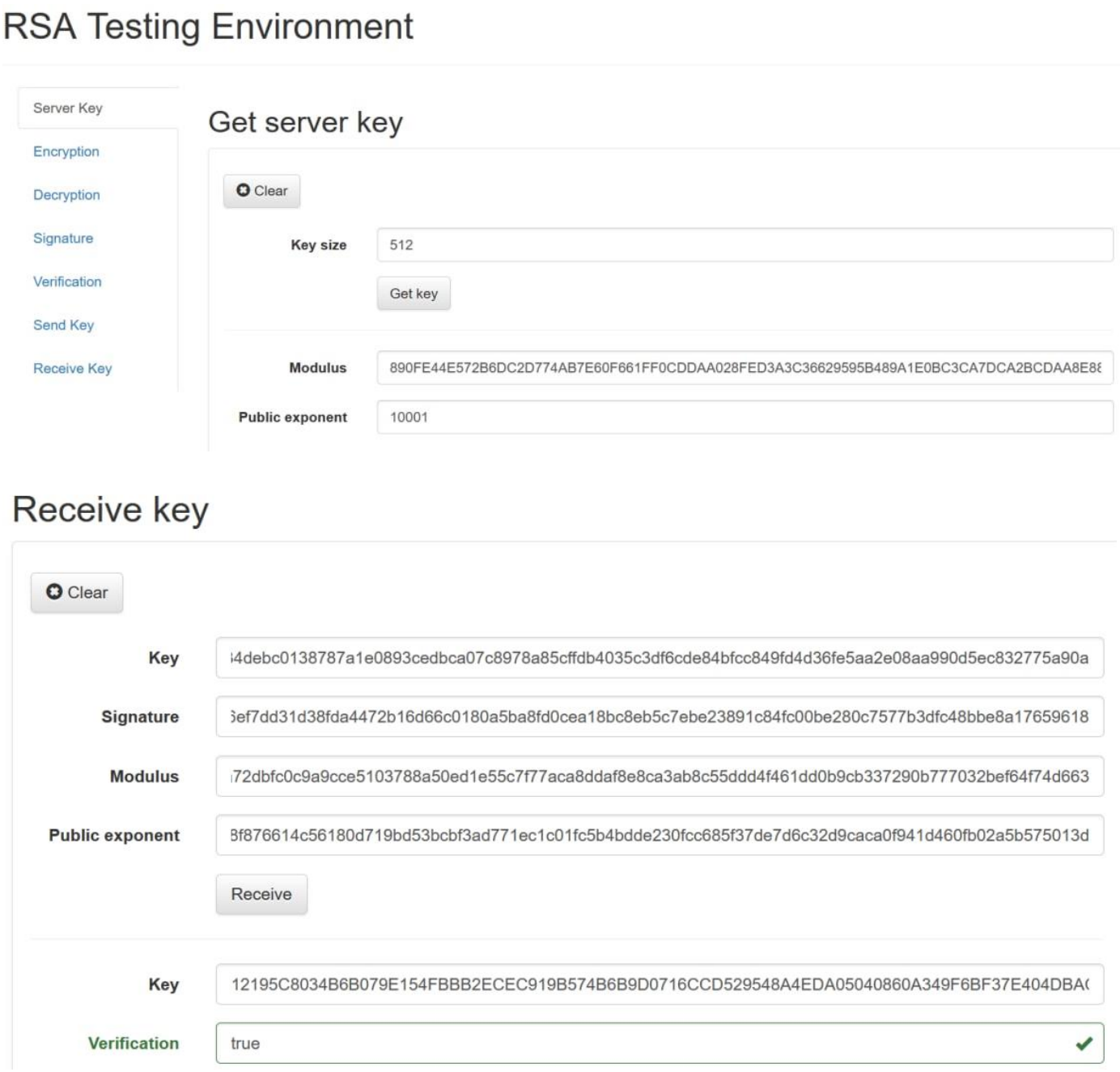
E =

24cc322e781b20ed64bcbb851166e66f8f876614c56180d719bd53bcbf3ad771ec1c01f
c5b4bdde230fcc685f37de7d6c32d9caca0f941d460fb02a5b575013d

S :

N =
890FE44E572B6DC2D774AB7E60F661FF0CDDAA028FED3A3C36629595B489A1E0BC3CA7DCA2BCDAA8E88CF590478C4D742F77126D22BC2C746B9A5AC442596419

E = 10001



Висновки:

У ході виконання цього комп’ютерного практикуму ми створювали секретний зв’язок між абонентами та електронний підпис за допомогою криптосистеми RSA. А також підраховували необхідні дані за допомогою модульної арифметики та імовірнісний тест Міллера – Рабіна, ознайомились з методами генерації параметрів для асиметричних криптосистем. Застосовували перевірку чисел на простоту, методи генерації ключів для асиметричної криптосистеми типу RSA. Засвоїли практичний метод захисту інформації на основі криптосхеми RSA, використання засекреченого зв'язку й електронного підпису, протокол розсилання ключів.