

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

КРИПТОГРАФІЯ КОМП'ЮТЕРНИЙ ПРАКТИКУМ №4
Вивчення крипtosистеми RSA та алгоритму електронного
підпису; ознайомлення з методами генерації параметрів для
асиметричних крипtosистем

Виконали студенти:
Зго курсу групи ФБ-31
Ткач Олександр Олександрович
Томашевський Назар Геннадійович

Київ – 2025

Мета роботи:

Ознайомлення з тестами перевірки чисел на простоту і методами генерації ключів для асиметричної крипtosистеми типу RSA; практичне ознайомлення з системою захисту інформації на основі крипtosхеми 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 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 повернати 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 і 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 $0 < k < n$.

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()`.

Хід роботи:

Написали скрипт згідно вимог та провели наше дослідження

Спочатку проводиться тест на простоту 10 невеликих(32 біт) кандидатів. Перевіряється спочатку методом відсіювання, якщо не хватає то уже переходить до тесту Міллера-Рабіна.

Далі починається сама демонстрація RSA

Генерація ключів для А та В, та їхня перевірка на довільно згенерованому відкритому тексті

Потім перевірка цифрових підписів та правильності шифрації тексту

Ну і сам протокол обміну ключами

=====

ДЕМО ГЕНЕРАЦІЇ ПРОСТИХ ЧИСЕЛ (КАНДИДАТИ ТА ВІДСІВ)

=====

[1] Кандидат:

candidate: 2149804877 | Шістнадцяткове: 80236B4D

-> ВІДСІЯНО МАЛИМИ ПРОСТИМИ (ділиться на 11)

[2] Кандидат:

candidate: 2818597293 | Шістнадцяткове: A80061AD

-> ВІДСІЯНО МАЛИМИ ПРОСТИМИ (ділиться на 3)

[3] Кандидат:

candidate: 3982628355 | Шістнадцяткове: ED621603

-> ВІДСІЯНО МАЛИМИ ПРОСТИМИ (ділиться на 3)

[4] Кандидат:

candidate: 2274127923 | Шістнадцяткове: 878C7033

-> ВІДСІЯНО МАЛИМИ ПРОСТИМИ (ділиться на 3)

[5] Кандидат:

candidate: 2376354455 | Шістнадцяткове: 8DA44A97

-> ВІДСІЯНО МАЛИМИ ПРОСТИМИ (ділиться на 5)

[6] Кандидат:

candidate: 3869595175 | Шістнадцяткове: E6A55627

-> ВІДСІЯНО МАЛИМИ ПРОСТИМИ (ділиться на 5)

[7] Кандидат:

candidate: 3226920265 | Шістнадцяткове: C056E549

-> ВІДСІЯНО МАЛИМИ ПРОСТИМИ (ділиться на 5)

[8] Кандидат:

candidate: 3325738403 | Шістнадцяткове: C63ABDA3

-> ВІДСІЯНО МАЛИМИ ПРОСТИМИ (ділиться на 13)

[9] Кандидат:

candidate: 2330433463 | Шістнадцяткове: 8AE797B7

-> ВІДСІЯНО МАЛИМИ ПРОСТИМИ (ділиться на 31)

[10] Кандидат:

candidate: 2244024657 | Шістнадцяткове: 85C11951

-> ВІДСІЯНО МАЛИМИ ПРОСТИМИ (ділиться на 3)

ПІДСУМОК: знайдено простих серед кандидатів: 0

ДЕМОНСТРАЦІЯ RSA

[1] Генерація ключових пар А і В...

Ключі абонента А:

p(A):

7866726441494819621179160349637674593916380052415597784216
0186757912656490749 | Шістнадцяткове:
ADEC180241BD5B74D809802086D83902389FF6907771F2709A62859
AC4A230FD

q(A):

9771191802900190300507056184448857193370031290641875892819
4441350104604012131 | Шістнадцяткове:
D806FC597C840DEFADB2F1B2F14DB894B759FEBA6FBB9FACDF0D
3984D248EE63

n(A) = p(A)*q(A):

768672929207923649473105835983855536157304640503676697181
622458729989532016946433880292186186383837549173171064028
1083550490444646089419150946744285276119 | Шістнадцяткове:
92C3F33BAAD3F3EF47872895B2FA072B86758156DF90FD0C611C87
D9476874D1FD7185BCCFACB555C268C227505F4C781F87EF4F28A
C59DC35867EEA256C27D7

φ(n(A)):

768672929207923649473105835983855536157304640503676697181
6224587299895320169287959620477911764621513326390845322408
219437059869909319064522838727024773240 | Шістнадцяткове:
92C3F33BAAD3F3EF47872895B2FA072B86758156DF90FD0C611C87
D9476874D0777E7161116B4BF13CAC5053D8395AE12F8DFA04417E
C7BEBC16BFCA8E810878

e(A): 65537 | Шістнадцяткове: 10001

d(A):

6988628787838035102896838615911322172503013107772949021402
715138536464327819201719242354339876634158895784038307150
064164627803356070257405638731552727673 | Шістнадцяткове:
856FB4B8BE1320C89F6ECA3E78D00C2ECD7C0E512EC8C2D93B7D
D6217D640238FB3BB1D7C01F9E974143A570E5B917E07EFD05A052
EFF2A97E96EC5F3C7B2279

Бітова довжина p(A): 256 | q(A): 256 | n(A): 512

Ключі абонента B:

p(B):

786585136737214574938983221499957465028180950923731808037

71920659881724379139 | Шістнадцяткове:

ADE7241B3998770471F538643AD68F90E2ECB9EECC9CF61A48278
ADB872E8003

q(B):

1089174055786536995705037979079249785931947980635511871211

91998529111789534119 | Шістнадцяткове:

F0CD1122E4809FCF4BE46500B731CE70E4CEED6396B58F4EBEEC6
A9D9E55FBA7

n(B) = p(B)*q(B):

856728123601479778306918984559763147614261537906615278756

924810966998802143370541793653179446237085899958037158309

8467239155955232099135308517313332343541 | Шістнадцяткове:

A393FF957D1DE565ED55CF7616778E8148E9EEBC66FB8E7F4DFB4
8C57A8DBE50AC9FDB26ADA3E697EE671CE21109964F002FF55A4D
328B1DD48D280921D772F5

$\phi(n(B))$:

856728123601479778306918984559763147614261537906615278756

924810966998802143351784201727941930530645687952245085800

2454346000030864174171389328319818430284 | Шістнадцяткове:

A393FF957D1DE565ED55CF7616778E8148E9EEBC66FB8E7F4DFB4
8C57A8DBE4F0DEBA5E88F8ACFC4308D7F7D1F01384D38744E07E9
E005B4CD79328FFC52F74C

e(B): 65537 | Шістнадцяткове: 10001

d(B):

1133380049685647752860367059238772981646344436524460147217
989549580218748887324712609515427397912736029196631657519
893787933842983236798540450074504871913 | Шістнадцяткове:
15A3D8A6101D4A61C79A188E08BEDC4D6236006D503A6B49902974
211BEB7087C8D1A9EB30422EE6477E10A59EDA6C765AC7982317D
F9921A825267941EA3FE9

Бітова довжина p(B): 256 | q(B): 256 | n(B): 512

[2] Вибір випадкового відкритого повідомлення M

Випадкове повідомлення M:

4650522273850019341528667249659018993519095836378008711669
581922855524207138122554174497981112659283163112825570713
30685147177013639596491647713458126201 | Шістнадцяткове:
58CB478485058BC4132CF96F83E5CDA3322245E1D7ADB33D7B7FC
4D3B29D39D64B4127334E50D9FE7A108B78F3B33054F28A563E729
10BAF81B96FB84E7DFD79

[3] Шифрування та розшифрування для A і B

C_A = M^e(A) mod n(A):

5896393701135937639243989611428532949872328154123505585282
326348322080163960724724773960668905900863261404532314621
574793176106794982409097093765050378720 | Шістнадцяткове:
7094F8D8410F5A373F6D7ADB50390B85EDFDD4769C929E70498217

0A0A92C4F144663FB2268337BDAB110C0737E6B956D10CF3B099AE
FB6B9B8453F1A23FA9E0

M_dec_A = C_A^d(A) mod n(A):

4650522273850019341528667249659018993519095836378008711669
5819228555242071381225554174497981112659283163112825570713
30685147177013639596491647713458126201 | Шістнадцяткове:
58CB478485058BC4132CF96F83E5CDA3322245E1D7ADB33D7B7FC
4D3B29D39D64B4127334E50D9FE7A108B78F3B33054F28A563E729
10BAF81B96FB84E7DFD79

Перевірка для A: OK

C_B = M^e(B) mod n(B):

491316260989599391775312510437340354867370241722943894853
027248480714125712363222539893147068101406528476532155687
5806875458627911134122889318035204860 | Шістнадцяткове:
1803D9919E89CE8CE71F305512BFB21C35E37B733F528D3B98AD88
0E5FBA31A9D2EB8A71490B5174E273D2B20A795D7C402C6D8C868
A706ABC522900E5AAFC

M_dec_B = C_B^d(B) mod n(B):

4650522273850019341528667249659018993519095836378008711669
5819228555242071381225554174497981112659283163112825570713
30685147177013639596491647713458126201 | Шістнадцяткове:
58CB478485058BC4132CF96F83E5CDA3322245E1D7ADB33D7B7FC
4D3B29D39D64B4127334E50D9FE7A108B78F3B33054F28A563E729
10BAF81B96FB84E7DFD79

Перевірка для B: OK

[4] Цифрові підписи абонентів А та В

Підпис $S_A = M^d(A) \text{ mod } n(A)$:

1487075036064168721315285238711677121402683654821644202161
6247111350273316390626047503840288260524425096555207494164
48589056068493868464209449499134772066 | Шістнадцяткове:
1C64ABDE8E3FB4B7DC9352A3931077B22706FBFD4AFB168485DDF
A663010B5AC3564E87930B765805B6A543D1BC5C130A38C82D271C
73631A508ACC89B0FC362

Перевірка підпису А: ПІДТВЕРДЖЕНО

Підпис $S_B = M^d(B) \text{ mod } n(B)$:

1222116620929368151348656492420847951100702506001010918572
344781566688806125131604887898067921009829077248283063412
140568358185312002015595688464371172395 | Шістнадцяткове:
175594E45638559F43A9550945E016D33D77C97B2EE4A8809370AFF
DF9B220B75D1E8FA1A5FD0220F0CE94E0631701F854B780B365F25
1847197AFB87FC0442B

Перевірка підпису В: ПІДТВЕРДЖЕНО

[5] Шифрування текстового повідомлення для В

Вихідний текст: Лабораторна-Томашевський-Ткач

Числове представлення тексту:

592284559799537931579444420540930381701067058175209236666
064972146242737486523709636454221957003597803949498779915
300197490801647866247 | Шістнадцяткове:
D09BD0B0D0B1D0BED180D0B0D182D0BED180D0BDD0B02DD0A2D

0BED0BCD0B0D188D0B5D0B2D181D18CD0BAD0B8D0B92DD0A2D0
BAD0B0D187

Шифротекст C_text:

8204362872820076068025789261862458979118472064679275528486
720440978532563348719502212133296810368499960245517900591
090803509943400889864379539446346583163 | Шістнадцяткове:
9CA617224EAЕ7880A29000BF77047AD2F75CBD03880D4EECD21C9
BCEE943E88687C949ACCAB712B7E915B21E7BDB10D1AA83DE356
7FE3AA8B29FF37A08DC8C7B

Розшифроване значення M_text:

59228455979953793157944420540930381701067058175209236666
064972146242737486523709636454221957003597803949498779915
300197490801647866247 | Шістнадцяткове:
D09BD0B0D0B1D0BED180D0B0D182D0BED180D0BDD0B02DD0A2D
0BED0BCD0B0D188D0B5D0B2D181D18CD0BAD0B8D0B92DD0A2D0
BAD0B0D187

Декодований текст: Лабораторна-Томашевський-Ткач

[6] Протокол обміну сесійним ключем (A → B)

Згенерований сесійний ключ k:

4229112933279024642707838283244528196473975116080949079817
8036523948081937179426110413797670191343612341476365764208
97064606292947260673344991147049283904 | Шістнадцяткове:
50BF7931D06925BFB2CB7F56FAF22DB97C26166FED64F51AD42A0
F3F8CB5FF77E45501A6DD1C043461DD844790613B367BBD5A0C09
DDA7B315767F76ADEAF140

Відправлено зашифрований ключ k1 та підпис S1.

Підпис $S = k^d(A) \bmod n(A)$:

6341393092472184728067872905760565078699991151796991148268
325824687531597365013040510053125638122861408810977013708
536695488942208879200389091687663632789 | Шістнадцяткове:
79141571A7E9D33B6F0389E2A5903FDFB6528846FF8235D20AC7A2
01ECC782421C6FE6B0F97BAB5CD42A37F32AE4D2964648CB425E3
EC003B7FA42F6C1BF8195

$k_1 = k^e(B) \bmod n(B)$:

156195543103793763038482451968380176709812194774655759439
0937153619857196570713138384905464480938211859145504167822
876818501958567042446052934367747868582 | Шістнадцяткове:
1DD2ADDA39C60BD19AB7EF14191DEBAF35CB2C39097B0A016A7D
5D12B26DC50E69E9981A5E0A70BAD7E0F6D901C2CDB538CF4B0E
504BBEBACCC2EC5A177D87A6

$S_1 = S^e(B) \bmod n(B)$:

1796317927334966985262811823515154451248244093710376961561
093994840671646036575256999075219981340781523188274381810
884489972168257220506503138794271358472 | Шістнадцяткове:
224C3818F152C6DEA6A696942DB54C94F450DA6BF0165408439F8A
81938AE6E2E3690CC75D492DF9D86AA8A5F6BF495CAA622593C57
03EE09F449D815F6B3E08

$k' = k_1^d(B) \bmod n(B)$:

4229112933279024642707838283244528196473975116080949079817
8036523948081937179426110413797670191343612341476365764208
97064606292947260673344991147049283904 | Шістнадцяткове:
50BF7931D06925BFB2CB7F56FAF22DB97C26166FED64F51AD42A0
F3F8CB5FF77E45501A6DD1C043461DD844790613B367BBD5A0C09
DDA7B315767F76ADEAF140

$S' = S_1^d(B) \bmod n(B)$:

6341393092472184728067872905760565078699991151796991148268
325824687531597365013040510053125638122861408810977013708
536695488942208879200389091687663632789 | Шістнадцяткове:
79141571A7E9D33B6F0389E2A5903FDFB6528846FF8235D20AC7A2
01ECC782421C6FE6B0F97BAB5CD42A37F32AE4D2964648CB425E3
EC003B7FA42F6C1BF8195

Статус перевірки підпису A на ключі k': АУТЕНТИФІКАЦІЯ
ПРОЙДЕНА

=====
ПРОТОКОЛ УСПІШНО ВИКОНАНО: $k' = k$, підпис валідний
=====

Висновки:

Під час виконання лабораторної роботи наша команда ознайомилася з тестами перевірки чисел на простоту і методами генерації ключів для асиметричної крипtosистеми типу RSA, на практиці ознайомилися з системою захисту інформації на основі крипtosхеми RSA, організували з використанням цієї системи засекреченого зв'язку й електронного підпису, вивчили та реалізували протокол розсилання ключів.