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

КРИПТОГРАФІЯ
КОМП’ЮТЕРНИЙ ПРАКТИКУМ №4

Виконала: Студентка 3-го курсу
Групи ФБ-93
Пономаренко Олександра Сергіївна

Київ 2021

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

Мета:

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.

Завдання до виконання:

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 $pq \le p_1 q_1$; 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 $0 < k < n$.

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().

Виконання роботи:

1) В якості тесту перевірки на простоту використала тест Міллера-Рабіна із попередніми пробними діленнями (перевірка ділення на 3, 5, 7 та 11)

```
669646526662011696954542354340925762434025897822304399186540791830818169346972 43
odd
669646526662011696954542354340925762434025897822304399186540791830818169346972 43
d: 33482326333100584847727117717704628812170129489111521995932703959154090846734862 1 2^s: 2 s: 1 p-1: 669646526662011696954542354340925762434025897822304399186540791830818169346972 42
counter: 0 k: 10
x: 30359261107704362034732638943368429233249352034643525782725267495010677927090953
GSD( 669646526662011696954542354340925762434025897822304399186540791830818169346972 43 , 30359261107704362034732638943368429233249352034643525782725267495010677927090953 )
GSD( 30359261107704362034732638943368429233249352034643525782725267495010677927090953 , 624613045079244562598895754735571777690388571294338835320354419306046108051533 7 )
GSD( 624613045079244562598895754735571777690388571294338835320354419306046108051533 7 , 537473930453457953077680875394555812563380918286997236991109072276883360502960 5 )
GSD( 537473930453457953077680875394555812563380918286997236991109072276883360502960 5 , 871391146257866095212148793410159651270076530073415983292453470291627475485732 )
GSD( 871391146257866095212148793410159651270076530073415983292453470291627475485732 , 146392426987382959503915993484600218013350002429476470156369901019068752115213 )
GSD( 146392426987382959503915993484600218013350002429476470156369901019068752115213 , 139429011320951297692568825987158561203326517926033632510603965196283714909667 )
x^d modp: 577764452463222593353143477471436489322783064533424332724423792555949560778121 86
x^(d*2^i)modp: 577764452463222593353143477471436489322783064533424332724423792555949560778121 86
not strong pseudo1
669646526662011696954542354340925762434025897822304399186540791830818169346972 43 is not prime!
669646526662011696954542354340925762434025897822304399186540791830818169346972 45
5|n
669646526662011696954542354340925762434025897822304399186540791830818169346972 45 is not prime!
669646526662011696954542354340925762434025897822304399186540791830818169346972 47
3|n
669646526662011696954542354340925762434025897822304399186540791830818169346972 47 is not prime!
```

2)Після усіх перевірок (у тому числі: добуток перших чисел менший за добуток другої пари), отримали дві пари простих чисел:

```
GenerateKeyPair #1: 669646526662011696954542354340925762434025897822304399186540791830818169346973 63 and 613769923729847564085466937360630910646906406790961376025214712178325582692061 21
GenerateKeyPair #2: 972730037219716669867979839741197351628751205182522830486634837865104386610613 17 and 857712260560254735875718853213258879906084424242323639448265389357821641227466 83
n #1: 411008897595300252547451721433833887726933993452336269334857084069439244179612214132499041975922778879514916395608009319163657444297979845812330076654140215892 3
n #2: 834322479138583911553049422310968634981305424110677837591783570157904901295732621096953618870800432704948350362432084627404232671421306368620424947022592136151 1
fi(n): 411008897595300252547451721433833887726933993452336269334857084069439244179612201298334538057330168479421999380041278509840611311640227728257289985216619825544 0
fi(n1): 834322479138583911553049422310968634981305424110677837591783570157904901295732602792530641071086375267961420817869769279047938422956607019618152717623137553512
```

3)Побудували відкриті ключі та секретні d:

```
final e: 129343391271373212417885369801370719435220618814473321938607653969947022786548583642484724981705407546977466310424885327719100356590946914715913477359992675968 1
final e1: 682532832428351900954468828394646526994588076671723118637102934013724132761980746369356793334377505893490502000073046943100258667094369425394463664571291904 21
```

```
final d: 674879758142620041166416770853648169760428494888279018498989244505451783657993273216327353732552999985284525552021755201474734409741767557395239876058079971008 1
final d1: 195772240709800846706850392133652957494997627234281996825128787176784512344972289060498935547431484003460073770868766965784115225404613031365736428090289120109 1
```

4)Шифруємо кожну пару і одразу ж розшифровуємо. Бачимо, що програма працює коректно. Теж саме із ЦП: створюємо його і одразу ж перевіряємо:

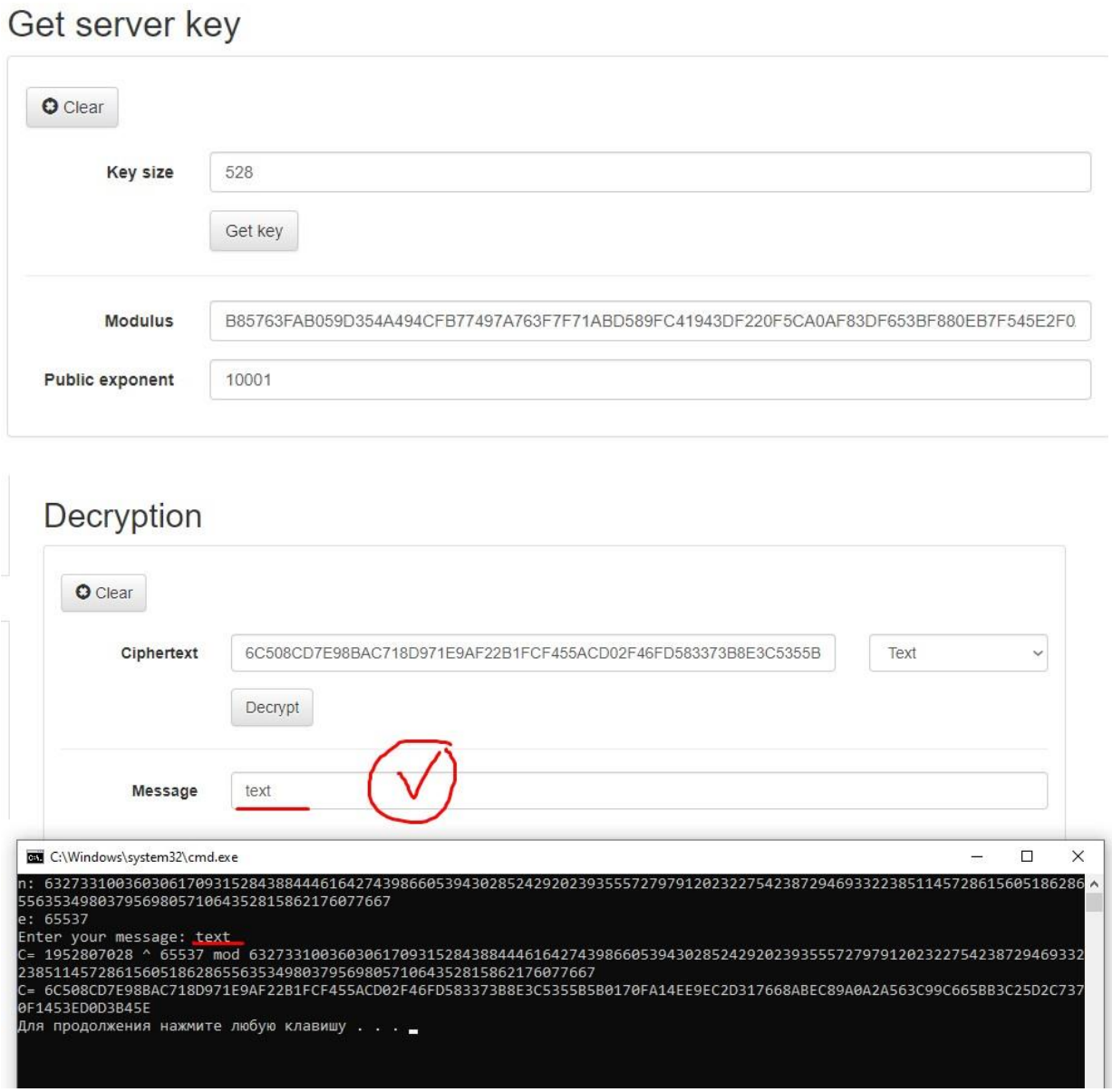
```
Enter your message: 23
23
C= 23 ^ 129343391271373212417885369801370719435220618814473321938607653969947022786548583642484724981705407546977466310424885327719100356590946914715913477359992675968 1
0766541402158923
C= 266852793573046269559954559333799177912708742791311238575587397625190054296626647617453381168190728776510668905911686160532523436149274017626938057383024490127 1
Enter your message: 11
11
C= 11 ^ 682532832428351900954468828394646526994588076671723118637102934013724132761980746369356793334377505893490502000073046943100258667094369425394463664571291904 21 mod 834322479138583911553049422310968634981305424110677837591783570157904901295732602792530641071086375267961420817869769279047938422956607019618152717623137553512
70225921361511
C1= 500395594740417109900172122522035122501623525511177170621649385361058839577449396050298681380497322601630156203984672517633483080864331296219466632021930141728 2
M= 266852793573046269559954559333799177912708742791311238575587397625190054296626647617453381168190728776510668905911686160532523436149274017626938057383024490127 1 ^ 674879758142620041166416770853648169760428494888279018498989244505451783657993273216327353732552999985284525552021755201474734409741767557395239876058079971008 1 mod 411008897595300252547451721433833887726933993452336269334857084069439244179612214132499041975922778879514916395608009319163657444297979845812330076654140215892 3
99710081
M= 23
M= 500395594740417109900172122522035122501623525511177170621649385361058839577449396050298681380497322601630156203984672517633483080864331296219466632021930141728 2 ^ 195772240709800846728
9120109 mod 834322479138583911553049422310968634981305424110677837591783570157904901295732621096953618870800432704948350362432084627404232671421306368620424947022592136151 1
M1= 11
S= 23 ^ 674879758142620041166416770853648169760428494888279018498989244505451783657993273216327353732552999985284525552021755201474734409741767557395239876058079971008 1 mod 411008897595300
766541402158923
S= 223244431935060469548460199603832034789107802505192839252668754204429055548516830536004114175934032817890476903007312110828642436384614713642783648337263661727 4
S= 11 ^ 195772240709800846706850392133652957494997627234281996825128787176784512344972289060498935547431484003460073770868766965784115225404613031365736428909028912010 9 mod 834322479138
947022592136151
S1= 681639209268410085250220105365578334799486114697354220819999070335373787322951579664283650681433405407386289196472092254030619842041127018987639616862271970820 4
M= 223244431935060469548460199603832034789107802505192839252668754204429055548516830536004114175934032817890476903007312110828642436384614713642783648337263661727 4 ^ 129343391271373212
926759681 mod 411008897595300252547451721433833887726933993452336269334857084069439244179612214132499041975922778879514916395608009319163657444297979845812330076654140215892 3
M= 23
M= 681639209268410085250220105365578334799486114697354220819999070335373787322951579664283650681433405407386289196472092254030619842041127018987639616862271970820 4 ^ 682532832428351900
9190421 mod 834322479138583911553049422310968634981305424110677837591783570157904901295732621096953618870800432704948350362432084627404232671421306368620424947022592136151 1
M1= 11
```

5)Створюємо протокол конфіденційного розсилання ключів з підтвердженням справжності по відкритому каналу за допомогою алгоритму RSA:

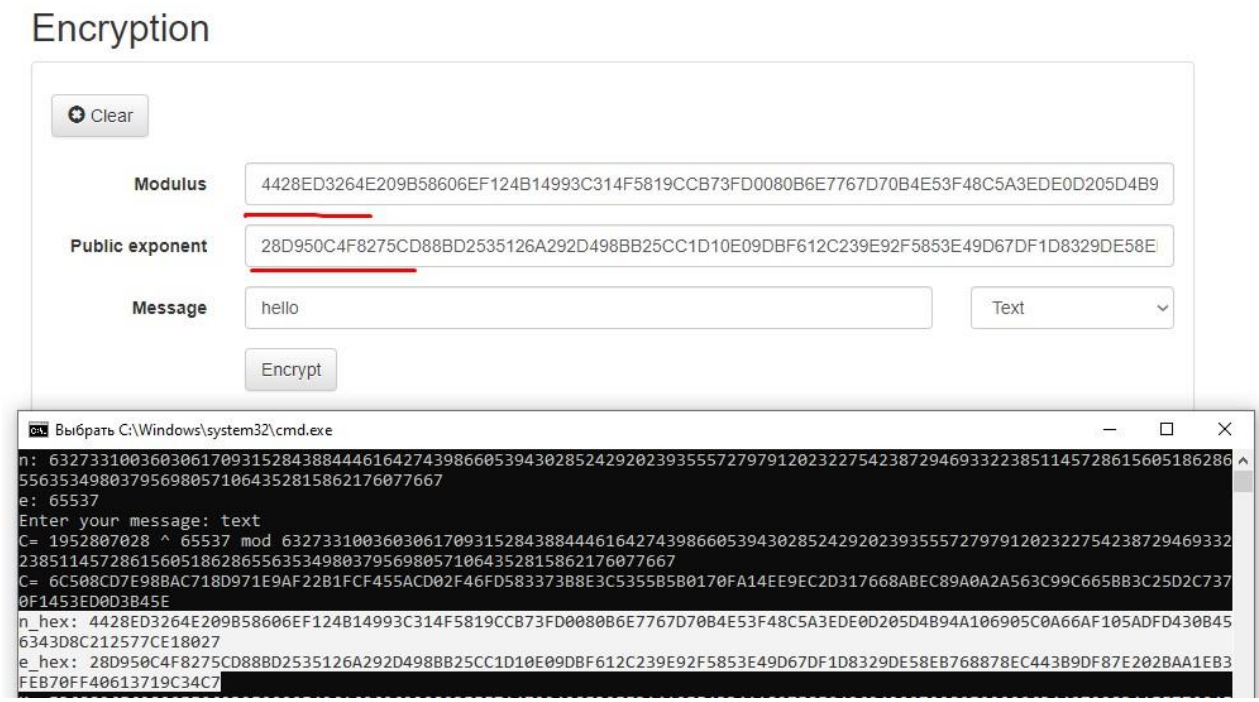
```
k: 338161237523618598862860853908262570501342018677344656248847959646598059825556782793139844122791725480898981667931144709692579608490278355434338791902092792492
k1= 338161237523618598862860853908262570501342018677344656248847959646598059825556782793139844122791725480898981667931144709692579608490278355434338791902092792492 ^ 682532832428351900954468828394646526994588
29190421 mod 834322479138583911553049422310968634981305424110677837591783570157904901295732621096953618870800432704948350362432084627404232671421306368620424947022592136151 1 = 135033139567069969303642405205623
2363370700012990729
S= 338161237523618598862860853908262570501342018677344656248847959646598059825556782793139844122791725480898981667931144709692579608490278355434338791902092792492 ^ 674879758142620041166416770853648169760428499
99710081 mod 411008897595300252547451721433833887726933993452336269334857084069439244179612214132499041975922778879514916395608009319163657444297979845812330076654140215892 3 = 285760499056095839190883823040363
6745012525808523730
S1= 285760499056095839190883823040363753009976782859521638595901534114811175413429240315635126723754916101492340596245351934946774700800276655676745012525808523730 ^ 682532832428351900954468828394646526994588
29190421 mod 834322479138583911553049422310968634981305424110677837591783570157904901295732621096953618870800432704948350362432084627404232671421306368620424947022592136151 1 = 589102835166397081031047262030267
8910941855053454900
A:
k1: 13503313956706996930642405205623258642356375852503885932652547632165875840779011745768961359551684088800758209481438703147051699006348643634236337070001299072 9
S1: 589102835166397081031047262030267009623842515034935431247251566237859916908815741112604495258809535775135679671852995713920864041130383602339891094185505345490 0
k= 13503313956706996930642405205623258642356375852503885932652547632165875840779011745768961359551684088800758209481438703147051699006348643634236337070001299072 9 ^ 195772240709800846706850392133652957494997
289120109 mod 834322479138583911553049422310968634981305424110677837591783570157904901295732621096953618870800432704948350362432084627404232671421306368620424947022592136151 1 = 338161237523618598862860853908262
4338791902092792492
S= 589102835166397081031047262030267009623842515034935431247251566237859916908815741112604495258809535775135679671852995713920864041130383602339891094185505345490 0 ^ 195772240709800846706850392133652957494997
289120109 mod 834322479138583911553049422310968634981305424110677837591783570157904901295732621096953618870800432704948350362432084627404232671421306368620424947022592136151 1 = 285760499056095839190883823040363
7674501252580852373 0
k(sign)= 285760499056095839190883823040363753009976782859521638595901534114811175413429240315635126723754916101492340596245351934946774700800276655676745012525808523730 ^ 129343391271373212417885369801370719
7735999926759681 mod 411008897595300252547451721433833887726933993452336269334857084069439244179612214132499041975922778879514916395608009319163657444297979845812330076654140215892 3 = 338161237523618598862860853
8355434338791902092792492
8:
k: 338161237523618598862860853908262570501342018677344656248847959646598059825556782793139844122791725480898981667931144709692579608490278355434338791902092792492
S: 285760499056095839190883823040363753009976782859521638595901534114811175413429240315635126723754916101492340596245351934946774700800276655676745012525808523730
k_1: 338161237523618598862860853908262570501342018677344656248847959646598059825556782793139844122791725480898981667931144709692579608490278355434338791902092792492
Press any key to continue . . .
```

6)Зробила перевірку на сайті:

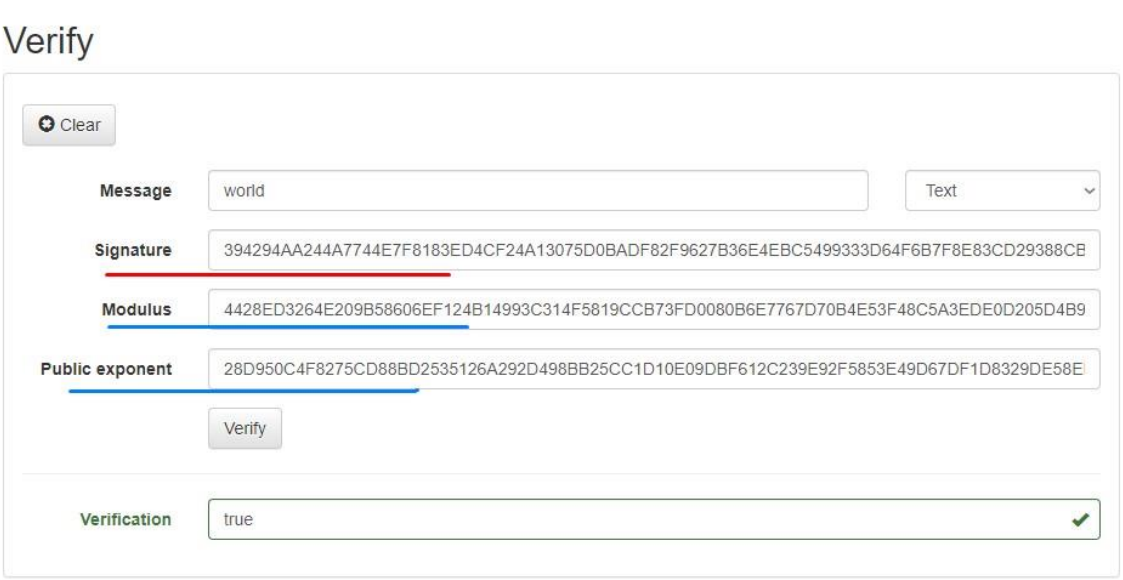
1.

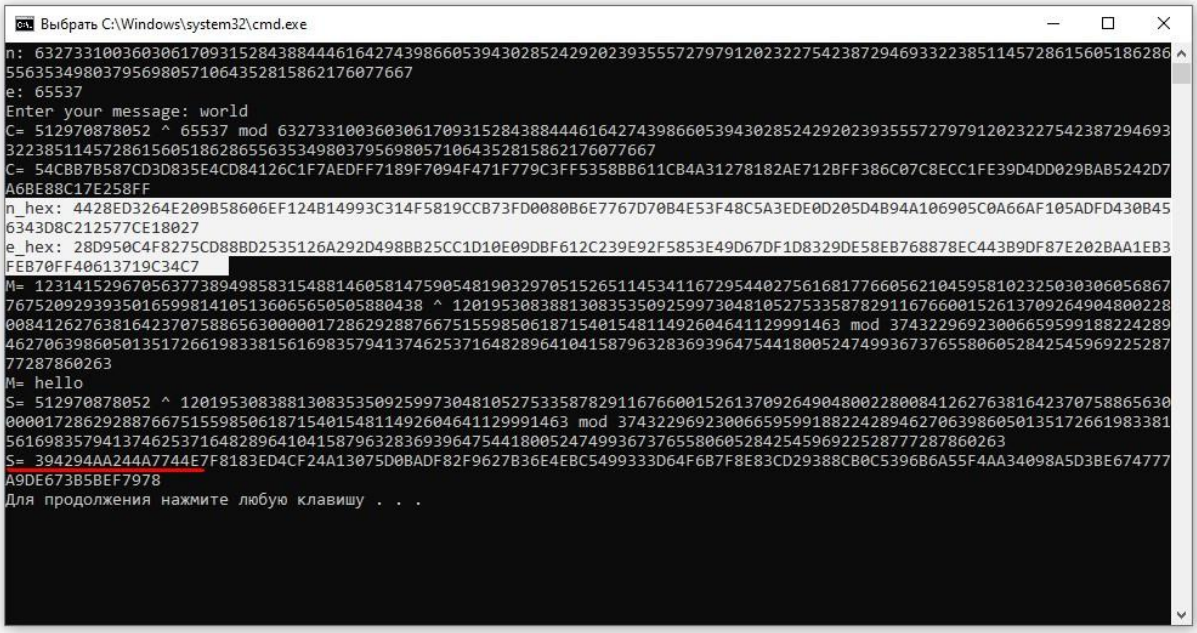


2.

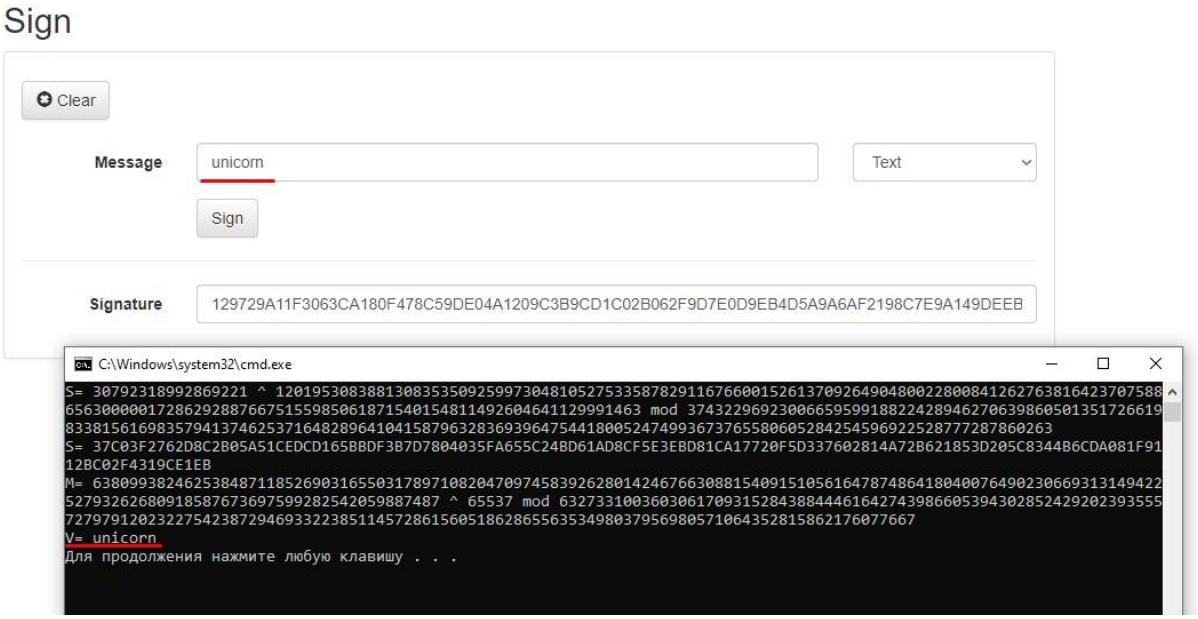

Encryption
Clear
Modulus
4428ED3264E209B58606EF124B14993C314F5819CCB73FD0080B6E7767D70B4E53F48C5A3EDE0D205D4B9
Public exponent
28D950C4F8275CD88BD2535126A292D498BB25CC1D10E09DBF612C239E92F5853E49D67DF1D8329DE58E
Message
hello
Text
Encrypt
Выбрать C:\Windows\system32\cmd.exe
n: 6327331003603061709315284388444616427439866053943028524292023935557279791202322754238729469332238511457286156051862865563534980379569805710643528158621760776 67
e: 65537
Enter your message: text
C= 1952807028 ^ 65537 mod 6327331003603061709315284388444616427439866053943028524292023935557279791202322754238729469332238511457286156051862865563534980379569805710643528158621760776 67
C= 6C508CD7E98BAC718D971E9AF22B1FCF455ACD02F46FD583373B8E3C5355B5B0170FA14EE9EC2D317668ABEC89A0A2A563C99C665BB3C25D2C7370F1453ED0D3B45E
n_hex: 4428ED3264E209B58606EF124B14993C314F5819CCB73FD0080B6E7767D70B4E53F48C5A3EDE0D205D4B94A106905C0A66AF105ADFD430B456343D8C212577CE18027
e_hex: 28D950C4F8275CD88BD2535126A292D498BB25CC1D10E09DBF612C239E92F5853E49D67DF1D8329DE58EB768878EC443B9DF87E202BAA1EB3FEB70FF40613719C34C7
Clear
Modulus
4428ED3264E209B58606EF124B14993C314F5819CCB73FD0080B6E7767D70B4E53F48C5A3EDE0D205D4B9
Public exponent
28D950C4F8275CD88BD2535126A292D498BB25CC1D10E09DBF612C239E92F5853E49D67DF1D8329DE58E
Message
hello
Text
Encrypt
Ciphertext
0166C3260D8FA65DB27B16233E5DECADBED421B3D93DD6F9F4AF6CC8FCAEE991B456DE8E32D4705A9FA
C:\Windows\system32\cmd.exe
n: 6327331003603061709315284388444616427439866053943028524292023935557279791202322754238729469332238511457286156051862865563534980379569805710643528158621760776 67
e: 65537
Enter your message: text
C= 1952807028 ^ 65537 mod 6327331003603061709315284388444616427439866053943028524292023935557279791202322754238729469332238511457286156051862865563534980379569805710643528158621760776 67
C= 6C508CD7E98BAC718D971E9AF22B1FCF455ACD02F46FD583373B8E3C5355B5B0170FA14EE9EC2D317668ABEC89A0A2A563C99C665BB3C25D2C7370F1453ED0D3B45E
n_hex: 4428ED3264E209B58606EF124B14993C314F5819CCB73FD0080B6E7767D70B4E53F48C5A3EDE0D205D4B94A106905C0A66AF105ADFD430B456343D8C212577CE18027
e_hex: 28D950C4F8275CD88BD2535126A292D498BB25CC1D10E09DBF612C239E92F5853E49D67DF1D8329DE58EB768878EC443B9DF87E202BAA1EB3FEB70FF40613719C34C7
M= 123141529670563773894985831548814605814759054810932970515265114534116729544027561681776605621045958102325030306056867675209293935016599814105136065650505880438 ^ 1201953083881308353509259973048105275335878291167660015261370926490480002280084126276381642370758865630000017286292887667515598506187154015481149260464112999146 3 mod 3743229692300665959918822428946270639860501351726619833815616983579413746253716482896410462780639860501351726619833815616983579413746253716482896410415879632836939647544180052474993673655806052842545969225287772878602 63
M= hello
Для продолжения нажмите любую клавишу . . .
def encode(string):
return int(string.encode().hex(), 16)
def decode(int):
return bytearray.fromhex(hex(int)[2:]).decode()
def int_to_hex(int):
return hex(int)[2:].upper()
def hex_to_int(hex):
return int(hex, 16)
n = "B85763FAB059D354A494CFB77497A763F7F71ABD589FC41943DF220F5CA0AF83DF653BF880EB7F545E2F0A5DA1
n = hex_to_int(n)
print("n:", n)
e = "10001"
e = hex_to_int(e)
print("e:",e)
M = encode(input("Enter your message: "))
C = Encrypt(M, e, n)
C = int_to_hex(C)
print("C=",C)
n1 = 3743229692300665959918822428946270639860501351726619833815616983579413746253716482896410462780639860501351726619833815616983579413746253716482896410415879632836939647544180052474993673655806052842545969225287772878602 63
n_hex = int_to_hex(n1)
print("n_hex:",n_hex)
e1 = 2243354844171827904141908533598054827684957610696113515644157327440866424240202428781126 75
e_hex = int_to_hex(e1)
print("e_hex:",e_hex)
d1 = 1201953083881308353509259973048105275335878291167660015261370926490480022800841262763816427075886563000001728629288766751559850618715401548114926046411299914 63
C_encrypted = "0166C3260D8FA65DB27B16233E5DECADBED421B3D93DD6F9F4AF6CC8FCAEE991B456DE8E32D4705A
C_encrypted = hex_to_int(C_encrypted)
M_decrypted = Decrypt(C_encrypted, d1, n1)
M_decrypted = decode(M_decrypted)
print("M=",M_decrypted)
126 %
Проблемы не найдены.
Стр: 39
Симв: 94
Вывод
Показать выходные данные из: Сборка

3.


Verify
Clear
Message
world
Text
Signature
394294AA244A7744E7F8183ED4CF24A13075D0BADF82F9627B36E4EBC5499333D64F6B7F8E83CD29388CB
Modulus
4428ED3264E209B58606EF124B14993C314F5819CCB73FD0080B6E7767D70B4E53F48C5A3EDE0D205D4B9
Public exponent
28D950C4F8275CD88BD2535126A292D498BB25CC1D10E09DBF612C239E92F5853E49D67DF1D8329DE58E
Verify
Verification
true


Выбрать C:\Windows\system32\cmd.exe
n: 6327331003603061709315284388444616427439866053943028524292023935557279791202322754238729469332238511457286156051862865
5563534980379569805710643528158621760776 67
e: 65537
Enter your message: world
C= 512970878052 ^ 65537 mod 6327331003603061709315284388444616427439866053943028524292023935557279791202322754238729469332238511457286156051862865563534980379569805710643528158621760776 67
C= 54CBB7B587CD3D835E4CD84126C1F7AEDFF7189F7094F471F779C3FF5358BB611CB4A31278182AE712BFF386C07C8ECC1FE39D4DD029BAB5242D7A6BE88C17E258FF
n_hex: 4428ED3264E209B58606EF124B14993C314F5819CCB73FD0080B6E7767D70B4E53F48C5A3EDE0D205D4B94A106905C0A66AF105ADFD430B456343D8C212577CE18027
e_hex: 28D950C4F8275CD88BD2535126A292D498BB25CC1D10E09DBF612C239E92F5853E49D67DF1D8329DE58EB768878EC443B9DF87E202BAA1EB3FEB70FF40613719C34C7
M= 123141529670563773894985831548814605814759054819032970515265114534116729544027561681776605621045958102325030306056867767520929393501659981410513606565050588043 8 ^ 120195308388130835350925997304810527533587829116766001526137092649048002280084126276381642370758865630000017286292887667515598506187154015481149260464112999146 3 mod 3743229692300665959918822428946270639860501351726619833815616983579413746253716482896410415879632836939647544180052474993673765580605284254596922528777287860263
S= 512970878052 ^ 120195308388130835350925997304810527533587829116766001526137092649048002280084126276381642370758865630000017286292887667515598506187154015481149260464112999146 3 mod 3743229692300665959918822428946270639860501351726619833815616983579413746253716482896410415879632836939647544180052474993673765580605284254596922528777287860263
S= 394294AA244A7744E7F8183ED4CF24A13075D0BADF82F9627B36E4EBC5499333D64F6B7F8E83CD29388CB0C5396B6A55F4AA34098A5D3BE674777A9DE673B5BEF7978
Для продолжения нажмите любую клавишу . . .

4.



Висновок:

За цю лабораторну роботу ми дізналися більше про асиметричні криптосистеми, в особливості про RSA, практично навчилися оформляти ЕЦП та перевіряти їх, розробили систему розсилання ключів з підтвердженням справжності по відкритому каналу за допомогою даного алгоритму