КРИПТОГРАФІЯ

КОМП’ЮТЕРНИЙ ПРАКТИКУМ №4

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

ФБ-23 Моісеєнко Дмитро

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 A, 1 p і q1 – 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) , (,) 1 n1 e та секретні d і d1 .

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().

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,

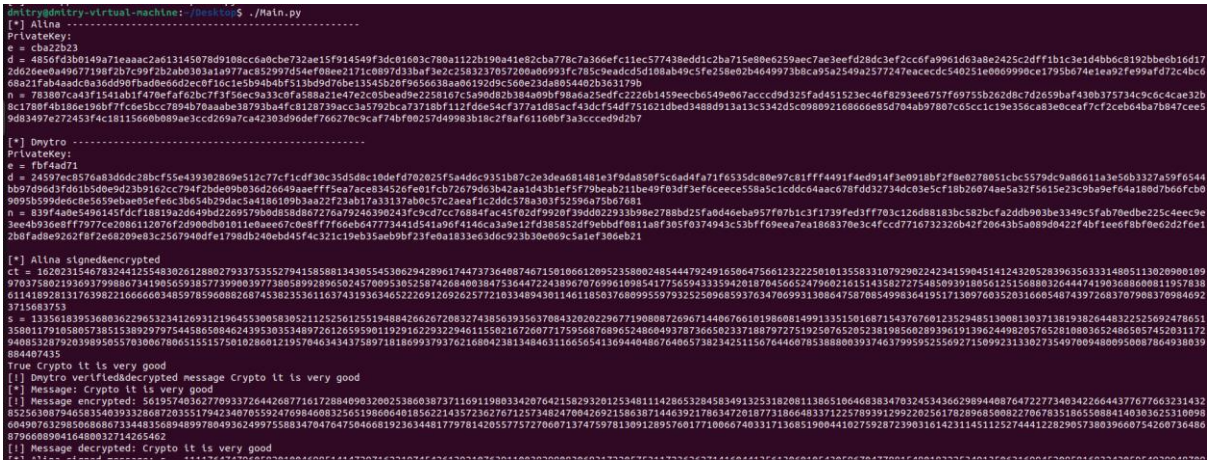
б) зашифрувати на сервері повідомлення для вашої реалізації та розшифрувати його локально.

Хід роботи:

Напишу консольний застосунок мовою Python для того, щоб був приватний і публічний ключа.

Результат виконання програми:

- *Створення ключів для довільних імен (Аліна та Дмитро)*
- *Шифрування та розшифрування тексту*
- *Передача зашифрованого повідомлення відкритим ключем з можливістю підтвердження особистості відправника*
- *Цифровий підпис*



1) *Зашифруємо повідомлення у себе використовуючи публічний ключ сервера та перевірка дешифрування на цьому сайті*

http://asymcryptwebservice.appspot.com/?section=rsa.

```
dmitry@dmitry-virtual-machine:~/Laba4_crypto$ ipython3
Python 3.10.12 (main, Nov  6 2024, 20:22:13) [GCC 11.4.0]
Type 'copyright', 'credits' or 'license' for more information
IPython 7.31.1 -- An enhanced Interactive Python. Type '?' for help.

In [1]: from RSA import *

In [2]: privateKey,publicKey = PrivKey.gen_RSA()

In [3]: message = "Hello from Client"

In [4]: serverPub = PubKey (0x1006,80a4899e51bf797396e0aaa7a99fdc63e78972349c44d
   ...: 304c6286278a69f8e358cdfe3e6a55c69c5793e02c692d46bc91175148991aa4427b8d26
   ...: 1a1933db018a2307c12f8ccdfe6a181630880f5f8d182d652c6dd60deacbf64d27bde3ab
   ...: 860431c87031dcdf05985f70e884d7807ff0e351bf7bb6a76aaac0ac327b55707181d9d9
   ...: bdc63bb2680e8d065a496dc3f925f37c5dc45960ea15a2d801a4fe7ea51559561b38a4d0
   ...: 42f5323b07f104af5e238cb384c53b22074fe927c1a52e7a66f7852fc0fae0a5e2528915
   ...: 28306be0c2392bb47923eccd2141dd1524c3ca9d52b3b1a511d4b6316cf69851674c4b57
   ...: 49677de8982f3496749257f0e9f115e1d91)
```

```
   ...: 749677de8982f3496749257f0e9f115e1d91)

In [7]: serverPub.encrypt(message)
Out[7]: 81316769931736817895372106385332723055905318997232739889656198328901653549326457955448188286884339796942560712170050486946354549932722302958377552218272160804237603983403878815512481545859225900945475476735959097855497437522201972503538304224503769755233869148428143791912936943901674315146897646652089471768249361681086295059910352369144152246826748538079788736326938459265596116690534218948245895345350621935096350435093441725564774830621243117153946231722645457591449650872695761811672685521191658700603972094131870334168336223648949505349345650142362864698009805301558457472996133895346208056560709734715491401 65

In [8]: hex(_)
Out[8]: '0x406a506e00abd5a0eb6b0a5b7bf07af4d8bdcfccee61016689a56f731f1caca6fe763a594ad3c89d571135420e1f1ea6dde3730ca8de7f624e7973d00bd57945c604ce012a1629a44f19db201e0d881ea6865c6672d62da269ce1bf7e63c12e4c541bf04fecc8cb6b0f190def5c266cfc735c67bcea3981e36e4aa93ae06b2b8dc3efdc3c53180492961820b35fe34a99cdb7c69cd7e319a3e3660b108f5115b37effa4f2da0112a5833e3ed30f48b175dd5ec09d60b9da5f10de597453ba74d58ab265cdef461ace4ea6d13b3b79260e59f4ea8bba293a69b4d93db527eb66759af641a7f1328882975db048d3c24b8e88c1ec71cdaad6a8554b7342989acc5'

In [9]: publicKey.export()
Out[9]:
[2264852039,
 19281806092352168627319457863632216409168213258645343529563061861238898913077726133279893811838084937930930795907149995084066661522589455785662280240042307787808136816543081777734825418882711706446535243364801708061215742792277442432300121290404393667563550981537313000234965512367705390272777859693379207629241657607936081270375591575462418180978767420245682098716056979724370577371032588763475970846136034170741986002147800389450916681135126218038772419856510473173126920143344539345940534807921345866171225434227134103976175556793454814393691954001616653508246517467304937732104245063379155376189313671774332658297]

In [10]: for i in _:
    ...:     print(hex(i)[2:])
    ...:
86fee647
98bdc78a685dd9c4b24b5dd7e7127456e6a2644d9518d72b3559e347d1da53b8d26f57d872e1207b5c4676167fc5056e4c9feafc732380e66e0d9aafd96e31dcc767ab780bf8b0827896c9549017dbf0aabc2fd38711acc328bc7173058c91b437a29f8393ca274d7bce46436b1ad9022ff305837250911f373ca173401133d612a2e79724ffb522028cc379c169d696622f028a4b9ea791222277e19faa2ef9ab12de66f9ea4984a40f28ab2ce661ae01c20fbd9d75cb700deb7ed5b9918274cf86d2afb064f17ffb403deee6fae0e731fdca365df3fb12d5d4b1761c68798cdfedb33677d3565ca72af03a1530229b4d8fb96a6ff2cd3b7be793d65ffe2279

In [11]: privateKey.decrypt(0x5E061022C732B9073E367A6CF0E7632505813C103192BFAA598CD17737A7D43978B10EE9B79EC2E2510DB674281BAAB62EEAB4
    ...: 40760B89EB8984356939873DE524E1DEC112A9F38752741B3889467661D332196C15C852841317DACFB157AF921B0A2E54DD3DFE0E472946A7E5DAF91F7
    ...: AE3FE159A93CC5C6FCD57147AF33363B7F7BB59582B83B1B39B0258BD0157195C9A7DBCD40F69154BDEF61C7B3B8AA54BAF49478D23D2A339AF10774612
    ...: 4E6ED1C0EEFE2C92CFC0B16F5CA348D39E99DFB498D9AFD834B72EF0E5549D9144440A802686237DDE74482D6C3F8759271AA6F8CDA17A93C7E6B3236AA
    ...: B5A74950650E4AEE0D4F578A57927E4A6BD1971D7)
Out[11]: 'Hello from Server'
```

Отримали значення – Hello from Server

RSA Testing Environment

- Server Key
- Encryption
- Decryption
- Signature
- Verification
- Send Key
- Receive Key

Encryption

Clear

Modulus 98bdc78a685dd9c4b24b5dd7e7127456e6a2644d9518d72b3559e347d1da53b8d26f57d872e1207b5c4676167fc5(

Public exponent 86fee647

Message Hello from Server — Text

Encrypt

Ciphertext 5E061022C732B9073E367A6CF0E7632505813C103192BFAA598CD17737A7D43978B10EE9B79EC2E2510DB(

```
dmitry@dmitry-virtual-machine:~/Laba4_crypto$ ./Main.py
[*] Alina ----------------------------------------------------
PrivateKey:
e = ba449fd1
d = 39da09cfcb47cf9c9f8169d62c753680c1f44f3259abdba7cd4561c1861db72d8db850cc90cf5ddfefa76bff0c778a588152480f9d1859e2bafc6868296ef271
52ec953c6df03fc6e4c1846876e631e6c490cf7014ffe251a86142a1b75ce2a5e3baf782d2b144ba972ada2b3caad9b5d17192c7afa942cf8064853242e3c0563a0b
3af8ff8fb5013dd88b3c14c13bbcbc46d3dc58a50bb1cb96faf472b8c37f05818b38cda450438420f11f2c0e0087269c6ad4fe58a13465cbde4e02af644958e7fbcc
89dda5cff6850ce25082319a053358e76105a959109f119a7c6bc756b3cedd54cda647f8dd82b6a220b381c4bbac18f148668eca0328ae5b93f26c61
n = 871891fbcc859af175abecee9ff8902d5473b015851d5ecb048398b0add3459a4d9eb8141dd7ad29645d7675548d2df419ff8f93afd0a9148407578ee666240b
c2ee469c2ca62bb412fab2a0f002d9b3e5cfec32a6083b6486edce47e6fe4f501d6fff69850ff4900c77213ad750449b343d72ff05210d82b4d055f9fce5c9cd21d5
c125731c976768822c8c5cf178d61b9a651bd80f504ffee7ec19182bdb1969bcc1b870159d9a9a01ae8d578533048a94e6396dccb2774254fdaa6fca5d2b4c711107
ed870ef49de4f3be8cc5efeb1b55608d30ed69d9c8594bb81fcb0e81c0c467713934caae140f15be6ca495f2bd5b906f15a3ab84ca009aa0cf54fcf1

[*] Dmytro ---------------------------------------------------
PrivateKey:
e = a892ce2b
d = 52e54b11d94ec8b2022b17f43796bdbc0a308323ab6b4ee3b2cabc857f5cfbf9a58d857d0f832d6b41b81665c29f89101fe1719bea667b5a90f4fd00773f0c89
9287b140cf2e866ba314cf7f10ba6618c7d839752c3bef5e4bc03edbeb4a7fd016a3322f5c270194c5e982055eb7fe6812b665abfac947d0d1e686fa1bdfc42e490d
fbd8f63c00cf495ed9d3d8055a814e0d9e97845c34dd1a8bf51b83f3ca437974c9ba91d0e79266af740a83f4b189f8a105b9facff8ec1bd805db6656a8060d2eed95
7bd300b6c11714d2bb70b522d3849c8e2afa06ee5bc7a41887fb8b67e323a9202a042a432c4230ecefa1f0afc0d6a22e90c821f60b3d95960443e23b
n = aaf3cafb74efd44857b900df05eba9515e0301782aab801064d95b18f51632e54e7475845496a81ad562c8635e8de51fbe416bf578f1e6d64977823a18517b9c
31eb00c544e3c5ed9e234d0b6516e219e224ef756f100f870c7545f1154a4e82d53bfb3110256367a8e387790266b576dd75ffcd3fef546ad3708e2ebd2425f34f4c
b3d94246e0b19774a4d99e8007677bd23f8f57dca3fab4d3523235f02faa0218154934ed435bcb7275bd1b7fc620fa2247802d5d9f273196dede7a5f663a24f82073
3bc14f3555dcb94bfcba2f847cda53595f9ae96c6416793155a05cf4069fcbf5ccf36ac3430f7179b2ed010a40bf64a35c79924741ffe8b4def222a1

[*] Alina signed&encrypted
ct = 353470618072204672517730430863247407764788746497637408460444891292970412849604526573636254439483727496327176506096400940209259 0
991155097553562700625382597738815093443428049913064557957853290364056809514225415821076502611587113437263415347355106064304457003225
505218159306256234065035573212083746925756856909742258856508195823785443530622744298705960505341629954378479764707564480543588335613
506251165966305359241331168209913731427774474824214872526118009284864071386788798763899025644949260814692717908146922652486669147996
761124223429709749334390896432598059365097035337723924736044402367285440614619835277941609464
s = 103909568859850660615785363257252903687269024707217259229466376684758170342545610208733281521159056208293543571301365828805377525
062663355644668495141866684641584848685657258191095113296077548331080272206306933423474307069196518997814646285907967500312941141708
882607471029240042763325151027650988814589043518922198066886906270373233736609743064615594624949364112551050187201947686261533846
470955215287344661668814313586877509242106358876739736696466816799879206967789531596749287097889764734161547832759467335235849820501
061282532677144862247767531881234136793455971606970729924530151459015882043778544044068953088
True Crypto it is very good
[!] Dmytro verified&decrypted message Crypto it is very good
[*] Message: Crypto it is very good
[!] Message encrypted: 155496957164579357350955842212461109503641908713039087670690987247437156400203241239007150166382430066527301 3
```

```
[!] Dmytro verified&decrypted message Crypto it is very good
[*] Message: Crypto it is very good
[!] Message encrypted: 5256137684635994782052820298861153267801917231291264652872459916204920259164010282134802466653843185468426052
547263373685065280904680572576644199058756495235408125321504134453314742903553235647422172370712712246982735538176805331943426655856
957090416236879447339996830726032698142008061635780520892888459806632835646443726398273719556583897975113724451601875030911031827780
072144681618103437882925417451727460408418639808423206138327739282264888367852013309009951701609573232231916845143491240610487290616
9032079734968277547581885372801357626730328269228982342961490092854422780449046561751578076475072747406198 2184
[!] Message decrypted: Crypto it is very good
[*] Alina signed message: s = 12357105390101438806785748534342757855971408468907203792923545972332003395022370264591679670127136974 1
411677253403083845522681716002802390250949121783313084083351364911548514738698073305181636399303173530480060441058393208603962538051
972274190600509879085234864188201045255451848639247633472545988922060281179298571479429494197468202307384243257875136313972380650986
168036012542583232200754694247010018862581502769548259561341891143660366834272428678843795551623468835011306260829916475164547255838
69411096994879796504901447643179669671427738682509487991154068116600109462561115772902031219303839500623258127415915725
[!] Alina public key verified message, result: True
[?] This is sign/verify oracle, it works with hex format
Press 1 to sign
Press 2 to verify
Press 3 to exit
[*] # 1
[!] Please enter here your Private Key [d,n]:
[*] d = 39da09cfcb47cf9c9f8169d62c753680c1f44f3259abdba7cd4561c1861db72d8db850cc90cf5ddfefa76bff0c778a588152480f9d1859e2bafc6868296e
f27152ec953c6df03fc6e4c1846876e631e6c490cf7014ffe251a86142a1b75ce2a5e3baf782d2b144ba972ada2b3caad9b5d17192c7afa942cf8064853242e3c056
3a0b3af8ff8fb5013dd88b3c14c13bbcbc46d3dc58a50bb1cb96faf472b8c37f05818b38cda450438420f11f2c0e0087269c6ad4fe58a13465cbde4e02af644958e7
fbcc89dda5cff6850ce25082319a053358e76105a959109f119a7c6bc756b3cedd54cda647f8dd82b6a220b381c4bbac18f148668eca0328ae5b93f26c61
[*] n = 871891fbcc859af175abecee9ff8902d5473b015851d5ecb048398b0add3459a4d9eb8141dd7ad29645d7675548d2df419ff8f93afd0a9148407578ee666
240bc2ee469c2ca62bb412fab2a0f002d9b3e5cfec32a6083b6486edce47e6fe4f501d6fff69850ff4900c77213ad750449b343d72ff05210d82b4d055f9fce5c9cd
21d5c125731c976768822c8c5cf178d61b9a651bd80f504ffee7ec19182bdb1969bcc1b870159d9a9a01ae8d578533048a94e6396dccb2774254fdaa6fca5d2b4c71
1107ed870ef49de4f3be8cc5efeb1b55608d30ed69d9c8594bb81fcb0e81c0c467713934caae140f15be6ca495f2bd5b906f15a3ab84ca009aa0cf54fcf1
[!] Please enter here reciever Public Key [e,n]:
[*] e = 1006
[*] n = 80a4899e51bf797396e0aaa7a99fdc63e78972349c44d304c6286278a69f8e358cdfe3e6a55c69c5793e02c692d46bc91175148991aa4427b8d261a1933d
b018a2307c12f8ccdfe6a181630880f5f8d182d652c6dd60deacbf64d27bde3ab860431c87031dcdf05985f70e884d7807ff0e351bf7bb6a76aaac0ac327b5570718
1d9d9bdc63bb2680e8d065a496dc3f925f37c5dc45960ea15a2d801a4fe7ea51559561b38a4d042f5323b07f104af5e238cb384c53b22074fe927c1a52e7a66f7852
fc0fae0a5e252891528306be0c2392bb47923eccd2141dd1524c3ca9d52b3b1a511d4b6316cf69851674c4b5749677de8982f3496749257f0e9f115e1d91
[!] Please enter here decimal value you want to sign
[*] pt = dmitry
```

Висновок:

Під час виконання комп’ютерного практикуму оволодів знаннями роботи з RSA криптосистемою. Під час результатів були отримані програмі, що дозволяють генерувати стійкі ключі, шифрувати та дешифровувати повідомлення.