

КРИПТОГРАФІЯ

КОМП'ЮТЕРНИЙ ПРАКТИКУМ №4

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

ФБ-23 Моїсєєнко 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 $1 < p, q < 2^{256}$ довжини щонайменше 256 біт. При цьому пари чисел беруться так, щоб $p \neq q$; p і q – прості числа для побудови ключів абонента А, $1 < p < q < 2^{256}$ – 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) , (d, p, q) і 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 M і знайти криптограму для абонентів А і 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:

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

```
python3 virtual-machine1:~/bin$ ./Main.py
[*] Alina -----
PrivateKey:
e = cba22523
d = 4856f4d3b0149a71eaaac2a613145078d9108cc6a8cbe732ae15f914549f3dc01603c780a1122b190a41e82cba778c7a366efc1ec57743bedd1c2ba715e80e6259aec7ae3eef428dc3ef2cc6fa9961d63a8e2425c2dff1b1c3e1d4bb6c8192bbe6b16d17
20d2eeea09671198f2b7c99f2b2ab0301a977ac852997d54ef08ee2171c0897d33ba73e2c2583237057200a06993fc785c9eadcd5d108ab49c5fe258e02b4649973b8ca95a2549a2577247eacecd540251e0069999ce1795b674e1ea92fe99afd72c4bc0
68a21fab4aadca36dd9f0fba0e6dd2ec0f16c1e5b94b4bf513bd9d76be13545b20f9656638aa0619209c560e23da0854402b363179b
n = 73380fca3f1541bf1478efaf62bc7f3f56ec9a33c0fa388a21e4f2ce05bead9e22581675a90d023384a90f98da25dfc222081459eeeb6549e087accdd32f5ad45152ce46f8293ee6757f6975b5623d6c7d2659af430b375734c0c6c4cae32b
8c1780f4b186e196b77f7f0c5bc7894b78aabe38793ba4fc8128739acc3a5792bca73718b711f1d6e54c3737a1d85acf43dcf544f751621dbed34880913a13c5342d5c098092168606e85d704ab97807c05cc11c356ca8e0caef7cf2ce6b4ba7b847cee5
9d83497e272453f4c1811560b089a9c3cd269a7ca42303d96def766270c9af74b700257d49983b18c2f8af61160bf3a3cccd9d2b7

[*] Dmitry -----
PrivateKey:
e = fbf4ad71
d = 2459fec8576a83ddcd38bcf55e439302809e512c77cf1cd730c354dbcb1d6f7d70282f54ddc09351b07c2e3da681481a3f9da858f5c6ad4fa71f6535dc88e97c81ff4491f4ed914f3e0918bf2f8e278051cbc5579dc9a86611a3e50b3327a59f6544
b89789a43f0d15d0e4d3b01c2cc7294f2bde9b036d2669aef4a9ac83452f6e01fcb728796d304aa1d43b1ef5779ebab211be49f03df3ef6ccee358a51cddc64aac078fdd32734dc03e5cf1b6874ae5a32f501e2c39ba9ef64a10d07b66fcb0
9895b599dece8e5659ebae5efec3b545b29dac5a4186109b3aa22f3ab17a3317ab0c57c2aef1c2ddc578a30f52596a75b67681
n = 839f4ae549e145fcd18819a2d649bd2269579b0d58d86726a79246390243fc9cd7cc76884fac45f02d7992039d022933b98e2788bd25fa0d46eba957f07b1c3f1739fed3ff703c126d88183bc582bca2dddb903be3349c5fab70edbe225c4eeca9e
3ee4b930e8ff7977c20801120f7d2908db01011eaae07c0e8ff7f06eb6477341d541a96f4140ca3a9e12fd385852df9ebdbf0811a8f305f0374943c53bfff09eeaf7ea1808370e3c4fcd7716732320b42f20643b5a089d0422f4bf1e0f8bf0e2d2f0e1
208fdae9262f872e68209e3c25c7940df779db240eb45f4c321c19eb35aeb9b723f0ba1833e03dc923b30e0d9c5a1ef330e6b21

[*] Alina signedEncrypted
cs = 1820315467832412554830261288027933753552794158081343055453062942896174473736408746715010661209523580024854447924916506475661232250101355831079290224234159045141243205283963563314805113020900189
9703758021936937998073141980505938577399001977380589928065024570095305258742604003847516447224189670769961098541775659433359420187045665247960216151435827275485093918056125156880326444741903688600811957838
611418928131763982126666034859785960882687453823536116374319363465226912692625772103348943011461185037680995597325250968593763478699313086475870854998364195171309760352031606540743972683707908370984692
3715083753
s = 133561839536803622965323412693112964553065830521125256125519488426627208327438563935637084320202967719080872606714406766101986001409133515816871543767601235294851300813037136193826448322525692478651
35801179185805738515309297975445865084624395303534897261265959011929162293229461155021672077175956876896524808493787360502337188797275192507625052381985602893961913902449820576520108036524865057452031172
940853287920398950557030067006515515750102860121957046343758971818099379376216804238134846131665654136944048676406573823425115676460785388000393746379959525569271509923113027354970094800950087864938039
884407435
True Crypto it is very good
[*] Dmitry verifiedDecrypted message Crypto it is very good
[*] Message: Crypto it is very good
[*] Message encrypted: 56195740362778931764426077161728040983200338063873716911908342076401587932012340114206532845834913253122001138051806468383470324533662989440876472773403426644777677663231433
8256338754608354019332668720351779423407859247696460832561980064018622143572362767125734824700426921586387144639217803472618773186648371225789391299228256170289945008277067835186550084140303625310098
604907632958606807344835689480970493624997588347047645046681923634481779781420557757270607137479578130912895760177100667403317136851900441027592873031612114511252744412282905738039600754260736486
87966889041640032714265462
[*] Message decrypted: Crypto it is very good
```

- 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
...: 304c6286278a69f8e358cdf3e6a55c69c5793e02c692d46bc91175148991aa4427b8d26
...: 1a1933db018a2307c12f8ccdf3e6a181630880f5f8d182d652c6dd60deacbf64d27bde3ab
...: 860431c87031dcdcf05985f70e884d7807ff0e351bf7bb6a76aac0ac327b55707181d9d9
...: bdc63bb2680e8d065a496dc3f925f37c5dc45960ea15a2d801a4fe7ea51559561b38a4d0
...: 42f5323b07f104af5e238cb384c53b22074fe927c1a52e7a66f7852fc0fae0a5e2528915
...: 28306be0c2392bb47923eccd2141dd1524c3ca9d52b3b1a511d4b6316cf69851674c4b57
...: 49677de8982f3496749257f0e9f115e1d91)
...: 749677de8982f3496749257f0e9f115e1d91)

In [7]: serverPub.encrypt(message)
Out[7]: 8131676993173681789537210638533272305590531899723273988965619832890165354932645795544818828688433979694256071217005048694635
454993272230295837755221827216080423760398340387881551248154585922590094547547673595909785549743752220197250353830422450376975523386
914842814379191293694390167431514689764665208947176824936168108629505991035236914415224682674853807978873632693845926559611669053421
894824589534535062193509635043509344172556477483062124311715394623172264545759144965087269576181167268552119165870060397209413187033
416833622364894950534934565014236286469800980530155845747299613389534620805656070973471549140165

In [8]: hex(_)
Out[8]: '0x406a506e00abd5a0eb6b0a5b7bf07af4d8bdcfccee61016689a56f731f1caca6fe763a594ad3c89d571135420e1f1ea6dde3730ca8de7f624e7973d00
bd57945c604ce012a1629a44f19db201e0d881ea6865c6672d62da269ce1bf7e63c12e4c541bf04fecc8cb6b0f190def5c266cfc735c67bcea3981e36e4aa93ae06b
2b8dc3efdc3c53180492961820b35fe34a99cdb7c69cd7e319a3e3660b108f5115b37effa4f2da0112a5833e3ed30f48b175dd5ec09d60b9da5f10de597453ba74d5
8ab265cdef461ace4ea6d13b3b79260e59f4ea8bba293a69b4d93db527eb66759af641a7f1328882975db048d3c24b8e88c1ec71cdaad6a8554b7342989acc5'

In [9]: publicKey.export()
Out[9]:
[2264852039,
 19281806092352168627319457863632216409168213258645343529563061861238898913077726133279893811838084937930930795907149995084066661522
5894557856622802400423077878008136816543081777734825418882711706446535243364801708061215742792277442432300121290404393667563550981537
313000234965512367705390272777859693379207629241657607936081270375591575462418180978767420245682098716056979724370577371032588763475
970846136034170741986002147800389450916681135126218038772419856510473173126920143344539345940534807921345866171225434227134103976175
556793454814393691954001616653508246517467304937732104245063379155376189313671774332658297]

In [10]: for i in _:
...:     print(hex(i)[:])
...:
86fee647
98bdc78a685dd9c4b24b5dd7e7127456ea2644d9518d72b3559e347d1da53b8d26f57d872e1207b5c4676167fc5056e4c9feafc732380e66e0d9aafd96e31dcc67
ab780bf8b0827896c9549017dbf0aabc2fd38711acc328bc7173058c91b437a29f8393ca274d7bce46436b1ad9022ff305837250911f373ca173401133d612a2e797
24ff522028cc379c169d696622f028a4b9ea791222277e19faa2ef9ab12de66f9ea4984a40f28ab2ce661ae01c20fbd9d75cb700deb7ed5b9918274c786d2afb064
f17ffb403deee6fae0e731fdca365df3fb12d5d4b1761c68798cdfed33677d3565ca72af03a1530229b4d8fb96a6ff2cd3b7be793d65ffe2279

In [11]: privateKey.decrypt(0x5E061022C73289073E367A6CF8E7632505813C103192BFAA598CD17737A7043978B10EE9879EC2E25100B6742B1BAAB62EEAB4
...: 407608089EB89843569398730E524E10EC112A9F38752741B3889467661D332196C15C852841317DACFB157AF92180A2E540D3DFE0E472946A7E5DAF91F7
...: AE3FE159A93CC5C6FCD57147AF33303B7F78B595828B381B3980258BD0157195C9A70BCD40F691548DEF61C7B3B8AA54BAF49478D23D2A339AF10774612
...: 4E0ED1C0EEFE2C92CFC0B10F5CA34BD39E99DFB498D9AFD834872EF0E5549D914440A802686237DDE74482D6C3F8759271AA6F8CDA17A93C7E6B3236AA
...: 85A7495065064AEE0D4F578A57927E4A68D1971D7)
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

98bdc78a685dd9c4b24b5dd7e7127456e6a2644d9518d72b3559e347d1da53b8d26f57d872e1207b5c4676167fc5c

Public exponent

86fee647

Message

Hello from Server

Text

Encrypt

Ciphertext

5E061022C732B9073E367A6CF0E7632505813C103192BFAA598CD17737A7D43978B10EE9B79EC2E2510DBc

```
dnitry@dnitry-virtual-machine:~/Laba4_crypto$ ./Main.py
[*] Alina -----
PrivateKey:
e = ba449fd1
d = 39da09cfcb47cf9c9f8169d62c753680c1f44f3259abdba7cd4561c1861db72d8db850cc90cf5ddfeafa76bfff0c778a588152480f9d1859e2bafcf6868296ef271
52ec953c6df03fc6e4c1846876e631e6c490cf7014ffe251a86142a1b75ce2a5e3baf782d2b144ba972ada2b3caad9b5d17192c7afa942cf8064853242e3c0563a0b
3af8ff8fb5013dd88b3c14c13bbcbcb46d3dc58a50bb1cb96faf472b8c37f05818b38cda450438420f11f2c0e0087269c6ad4fe58a13465cbde4e02af644958e7fbcc
89dda5cfff6850ce25082319a053358e76105a959109ff119a7c6bc756b3cedd54cda647f8dd82b6a220b381c4bbac18f148668eca0328ae5b93f26c61
n = 871891fbcc859af175abeceeff8902d5473b015851d5ecb048398b0add3459a4d9eb8141dd7ad29645d7675548d2df419ff8f93afd0a9148407578ee666240b
c2ee469c2ca62bb412fab2a0f002d9b3e5cfec32a6083b6486edce47e6fe4f501d6fff69850ff4900c77213ad750449b343d72ff05210d82b4d055f9fce5c9cd21d5
c125731c976768822c8c5cf178d61b9a651bd80f504ffee7ec19182bdb1969bcc1b870159d9a9a01ae8d578533048a94e6396dccbb2774254fdaa6fca5d2b4c711107
ed870ef49de4f3be8cc5efeb1b55608d30ed69d9c8594bb81fcb0e81c0c467713934caae140f15be6ca495f2bd5b906f15a3ab84ca009aa0cf54fcf1

[*] Dmytro -----
PrivateKey:
e = a892ce2b
d = 52e54b11d94ec8b2022b17f43796dbdc0a308323ab6b4ee3b2cabcf857f5cfbf9a58d857d0f832d6b41b81665c29f89101fe1719bea667b5a90f4fd00773f0c89
9287b140cf2e866ba314cf7f10ba6618c7d839752c3bef5e4bc03edbe47fd016a3322f5c270194c5e982055eb7fe6812b665abfac947d0d1e686fa1bdfc42e490d
fbd8f63c00cf495ed9d3d8055a814e0d9e97845c34dd1a8bf51b83f3ca437974c9ba91d0e79266af740a83f4b189f8a105b9facff8ec1bd805db6656a8060d2eed95
7bd300b6c11714d2bb70b522d3849c8e2afa06ee5bc7a41887fb8b67e323a9202a042a432c4230ecefaf1f0afcd06a22e90c821f60b3d95960443e23b
n = aaf3caf74ef44857b900df05eba9515e0301782aab801064a95b18f51632e54e7475845496a81ad562c8635e8de51fbae416bf578f1e6d64977823a18517b9c
31eb00c544e3c5ed9e234d0b6516e219e224ef756f100f870c7545f1154a4e82d53bf3110256367a8e387790266b576dd75ffcd3fe546ad3708e2ebd2425f34f4c
b3d94246e0b19774a4d99e8007677bd23f8f57dca3fab4d352325f02faa0218154934ed435bcb7275bd1b7fc620fa2247802d5d9f273196dede7a5f663a24f82073
3bc14f3555dcb94bfcb2a7847cda53595f9ae96c6416793155a05cf4069fcbf5ccf36ac3430f7179b2ed010a40bf64a35c79924741ffe8b4def222a1

[*] Alina signed&encrypted
ct = 3534706180722046725177304308632474077647887464976374084604448912929704128496045265736362544394837274963271765060964009402092590
991155097553562700625382597738815093443428049913064557957853290364056809514225415821076502611587113437263415347355106064304457003225
505218159306256234065035573212083746925756856909742258856508195823785443530622744298705960505341629954378479764707564480543588335613
50625116596630539241331168209913731427774474824214872526118009284864071386788798763899025644949260814692717908146922652486669147996
761124223429709749334390896432598059365097035337723924736044402367285440614619835277941609464
s = 10390956885985066061578536325725290368726902470721725922946637668475817034254561020873328152115905620829354357130136582880537752
50626633556466849514186668464158484868565725819109511329607754833108027220630693342347430706919651899781464628590796750031294114170
888260747102924004276332515102765098881458904351892219806688690627037323373660974306461559462494936411255105018720194768626812533846
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: 1554969571645793573509558422124611095036419087130390876706909872474371564002032412390071501663824300665273013
```



```
[!] Dmytro verified&decrypted message Crypto it is very good
[*] Message: Crypto it is very good
[!] Message encrypted: 5256137684635994782052820298861153267801917231291264652872459916204920259164010282134802466653843185468426052
547263373685065280904680572576644199058756495235408125321504134453314742903553235647422172370712712246982735538176805331943426655856
957090416236879447339996830726032698142008061635780520892888459806632835646443726398273719556583897975113724451601875030911031827780
072144681618103437882925417451727460408418639808423206138327739282264888367852013309009951701609573232231916845143491240610487290616
90320797349682775475818853728013576267303282692289823429614900928544227804490465617515780764750727474061982184
[!] Message decrypted: Crypto it is very good
[*] Alina signed message: s = 123571053901014388067857485343427578559714084689072037929235459723320033950223702645916796701271369741
411677253403083845522681716002802390250949121783313084083351364911548514738698073305181636399303173530480060441058393208603962538051
972274190600509879085234864188201045255451848639247633472545988922060281179298571479429494197468202307384243257875136313972380650986
16803601254258322200754694247010018862581502769548259561341891143660366834272428678843795551623468835011306260829916475164547255838
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 = 39da09cfcb47cf9cf9f8169d62c753680cf44f3259abdba7cd4561c1861db72d8db850cc90cf5ddffefa76bfff0c778a588152480f9d1859e2bafcf6868296e
f27152ec953c6df03fc6e4c1846876e631e6c490cf7014ffe251a86142a1b75ce2a5e3baf782d2b144ba972ada2b3caad9b5d17192c7afa942cf8064853242e3c056
3a0b3af8ff8fb5013dd88b3c14c13bbcb46d3dc58a50bb1cb96faf472b8c37f05818b38cda450438420f11f2c0e0087269c6ad4fe58a13465cbde4e02af644958e7
fbc89dda5cfff6850ce25082319a053358e76105a959109f119a7c6bc756b3cedd54cda647f8dd82b6a220b381c4bbac18f148668eca0328ae5b93f26c61
[*] n = 871891fbcc859af175abec9ef8902d5473b015851d5ecb048398b0add3459a4d9eb8141dd7ad29645d7675548d2df419ff8f93afd0a9148407578ee666
240bc2ee469c2ca62bb412fab2a0f002d9b3e5cfec32a6083b6486edce47e6fe4f501d6fff69850ff4900c77213ad750449b343d72ff05210d82b4d055f9fce5c9cd
21d5c125731c976768822c8c5cf178d61b9a651bd80f504ffee7ec19182bdb1969bcc1b870159d9a9a01ae8d578533048a94e6396dccb2774254fdaa6fca5d2b4c71
1107ed870ef49de4f3be8cc5efeb1b55608d30ed69d9c8594bb81fcb0e81c0c467713934caae140f15be6ca495f2bd5b906f15a3ab84ca009aa0cf54fcf1
[!] Please enter here reciever Public Key [e,n]:
[*] e = 1006
[*] n = 80a4899e51bf797396e0aaa7a99fdc63e78972349c44d304c6286278a69f8e358cdfce3e6a55c69c5793e02c692d46bc91175148991aa4427b8d261a1933d
b018a2307c12f8ccdf6a181360880f5f8d182d652c6dd60deacbf64d27bde3ab860431c87031dcdff05985f70e884d7807ff0e351bf7bb6a76aaac0ac327b5570718
1d9d9bdc63bb2680e8d065a496dc3f925f37c5dc45960ea15a2d801a4fe7ea51559561b38a4d042f5323b07f104af5e238cb384c53b22074fe927c1a52e7a66f7852
fc0fae0a5e252891528306be0c2392bb47923eccd2141dd1524c3ca9d52b3b1a511d4b6316cf69851674c4b5749677de8982f3496749257f0e9f115e1d91
[!] Please enter here decimal value you want to sign
[*] pt = dmytro
```

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.