Приложение 2. Демонстрация работы программы

Часть 1. Генератор ключей

In [1]:
```
# Сначала очистим директорию от артифактов предыдущих запусков кода

!rm -f encrypted.* decrypted.* *.public *.secret
!ls -l
```

```
total 1300
-rw-r--r-- 1 evgeny evgeny 431919 Apr 25 19:45 analyze.ipynb
-rw-r--r-- 1 evgeny evgeny  87080 Apr 25 17:06 demo.ipynb
-rw-r--r-- 1 evgeny evgeny  35117 Apr 23 16:57 image.png
-rw-r--r-- 1 evgeny evgeny   4172 Apr 26 17:00 keygen.py
-rw-r--r-- 1 evgeny evgeny  24721 Apr 29 10:25 manual.ipynb
-rw-r--r-- 1 evgeny evgeny 399380 Apr 29 10:27 manual.pdf
-rw-r--r-- 1 evgeny evgeny 306562 Apr 25 16:42 mayday.png
-rw-r--r-- 1 evgeny evgeny    408 Apr 20 15:36 open.txt
drwxr-xr-x 2 evgeny evgeny   4096 Apr 26 17:09 __pycache__
-rw-r--r-- 1 evgeny evgeny   5100 Apr 25 19:02 rsa.py
-rw-r--r-- 1 evgeny evgeny     13 Apr 25 17:06 secret.crack
drwxr-xr-x 6 evgeny evgeny   4096 Apr 15 15:18 venv
```

In [2]:
```
# Запускаем генерацию ключевой пары
# Скрипт по-умолчанию создаёт пару ключей с названиями "key.pub"
# и "key.secret", или можно напрямую передать скрипту префикс и
# получить пару ключей с другими названиями

# Генератор ключевой пары принимает два парметра p_size и q_size
# Чтобы соблюсти условие значительной величины их разницы -
# задам им разную длину

from keygen import KeyGenerator

KeyGenerator(1024, 3072)

# Проверим создались ли файлы с ключами
!ls -l
```

```
total 1308
-rw-r--r-- 1 evgeny evgeny 431919 Apr 25 19:45 analyze.ipynb
-rw-r--r-- 1 evgeny evgeny  87080 Apr 25 17:06 demo.ipynb
-rw-r--r-- 1 evgeny evgeny  35117 Apr 23 16:57 image.png
-rw-r--r-- 1 evgeny evgeny   4172 Apr 26 17:00 keygen.py
-rw-r--r-- 1 evgeny evgeny   1541 Apr 29 10:28 key.public
-rw-r--r-- 1 evgeny evgeny   2053 Apr 29 10:28 key.secret
-rw-r--r-- 1 evgeny evgeny  24721 Apr 29 10:25 manual.ipynb
-rw-r--r-- 1 evgeny evgeny 399380 Apr 29 10:27 manual.pdf
-rw-r--r-- 1 evgeny evgeny 306562 Apr 25 16:42 mayday.png
-rw-r--r-- 1 evgeny evgeny    408 Apr 20 15:36 open.txt
drwxr-xr-x 2 evgeny evgeny   4096 Apr 26 17:09 __pycache__
-rw-r--r-- 1 evgeny evgeny   5100 Apr 25 19:02 rsa.py
-rw-r--r-- 1 evgeny evgeny     13 Apr 25 17:06 secret.crack
drwxr-xr-x 6 evgeny evgeny   4096 Apr 15 15:18 venv
```

In [3]:
```
# Посмотрим содержимое key.pub
# Файл содержит строки в 16-м формате со значением
# e (открытый ключ) и n (модуль) разделенные знаком /

!cat key.public
```

```
0XA7733914AF645573B787DDD55F1CA35CF3FC0DA25639A97F784A343D4223A4E99
F784969A54D8D71FEA3BADB57B8C25ADC6F0367C30CA023E361D35DF5DE1243CE6F
B2FDC386D3B9F371D2F8B360584F7DABE642146A00C77E51E2BF802199D3FF3C892
D73A39B2F5C0BE9FB261C32D1E9F3186C98424B99748347FF0EAA1217F3D5EC49D1
EEB0CF0D9606309A1E778B2BAC6AD19E71D0A8B128C8D055C8FBF30662DD0B41BAA
DFFF8C5CB2F352D3154FAEF5B424DA62440C7A6C7786CD6ACB324479288E8CDA014
85AFCACE3DF0AD9824DBB6932CA32778A332BA09E5C88B9C621CB4D1A663276371B
3CE692AAE131729BE11BEA7BE545A8B0EB130CC966D3F/0X86C7A34204FEB43E5C3
B4940AD51819F0F729F892EF3B0F7274759D4C7BFB24B4940DC676DDEA556849E32
60288454B197CF90CEBE2FF14FD926B3DB72B5E374256294AE0E8DC2E015579DC15
2ED78AD54EC273194194812DF0368A2EE14C7FE9574452ACB23B477817BD961B47E
7101ED529A4FB771F842E042DF4F799835851B569F4FADC389940FEDBAAAF151213
EAE9EC51E2E0F6428005125C4EE3D5B06B64820752FA18DF042DB53963B85E8A8AC
462F14778FB37A5DA2DA5CAAA4157CE23CA44F3C52C50E21FC8737139C61609A0D2
A47093AB29F7386A617DB7CD0B15C988628601B243876D39459092CCBEE843A67BB
BB908935A97FEBEB50F033CAAE328D4C84A6F187B45EE6DF7FD4FF0CDDBF8E6378A
D308C4FD5F8C08648B77D75036CB9B034DEF6CDCAB27691917465F509B49E591DC5
91AC7A86B78C5C6D4AEF3A222617AD670DEA4EE22AD63FF41F948BB32B827D7ECE4
77F9866CD2E7DE7D083BE156A187CF4F0B18190952C2E3FBF8872815B09DD6635C0
F53417494354BDD47F1F95A6503BF237D430E39CE737F7D86340C96AEB9974E1C27
7EDBFE20C607BAF8C1020999468E09B004979C3A0AEAE773E814646E73E33900D6C
1D55FFF6DC40EF8D580617DAC0D72DF0A9B7D5F87EEF5D02AD046991AB742D63C35
B58CD59C3BED3385558762424EE8BEDA972026AA1263CAFE1B41754F4D70F5E25F3
```

In [4]:
```
# Посторим файл key.secret
# Файл содержит строки в 16-м формате со значением
# d (закрытый ключ) и n (модуль) разделенные знаком /

!cat key.secret
```

```
0X45470070F08F341787D931C8291287A4394B387086E8F29ED105E76FB058F59B8
6109BDC3B25EB0396E8C5F055B7CB29E26074C99227113CFEBCF6C009DEF7EED47E
82EAB95C840EFAFA7017A745CF31E7C3B637F4D720694EA2A4A52C345EAD99F6CC6
7BA0D1DAFF5FB86B02804D207AF26AA14960D3374C3F0938E630A193FBF4544CFDA
9E94015FE35D4FDC6F964229EB7A0F79AD762A91F1899641C5BA8036609B45A8A48
1D5F3DBEDAAB8972EB1A04345D0170803FC76C60AB2BC03C92A9E41EB26A80915C1
67545570F0B2E2FEB4C3B324545E318F93D8B5E909B9E86FE14BE16A918AFD7E3DD
B0277D8BFDBE6EE8D51BB5C20CA837C9AA7EE13536841A57C858C699752E3EE65B3
9625D88D6075F98F947B0B7895B74B777BAA9D5C5407496E3EDC84D201E8CBEE2F3
2B998DB78B1512FC42D9C3BE40CB29F0270E24617CEE20E3E04210F47FBF2F2785B
62D6037B39087AED14DE7ED322729545BA819C366F12F63B5D29415A86632D8416C
445F3CD52F8D4095ADDA03B5D96886F0EE9FCD4B7F058027A3438AA7A23894BCA74
3A625980F3128B1F7245C960EB70335FD4B2DDD1D7D9CE24B98DB21E254563288258
8E8D0EB1F2B62E8D4B55BFF7AB68BA5C7426D6A54CC9E4B589D3B1C54E36C26C384
D3E5162C77279625F0608EBFC47FB8755DFBF2CDFC36A16D11310CB049F87AE50DE
01CCA7A128E36F93E6A5F/0X86C7A34204FEB43E5C3B4940AD51819F0F729F892EF
3B0F7274759D4C7BFB24B4940DC676DDEA556849E3260288454B197CF90CEBE2FF1
4FD926B3DB72B5E374256294AE0E8DC2E015579DC152ED78AD54EC273194194812D
F0368A2EE14C7FE9574452ACB23B477817BD961B47E7101ED529A4FB771F842E042
DF4F799835851B569F4FADC389940FEDBAAAF151213EAE9EC51E2E0F6428005125C
4EE3D5B06B64820752FA18DF042DB53963B85E8A8AC462F14778FB37A5DA2DA5CAA
A4157CE23CA44F3C52C50E21FC8737139C61609A0D2A47093AB29F7386A617DB7CD
0B15C988628601B243876D39459092CCBEE843A67BBBB908935A97FEBEB50F033CA
AE328D4C84A6F187B45EE6DF7FD4FF0CDDBF8E6378AD308C4FD5F8C08648B77D750
36CB9B034DEF6CDCAB27691917465F509B49E591DC591AC7A86B78C5C6D4AEF3A22
2617AD670DEA4EE22AD63FF41F948BB32B827D7ECE477F9866CD2E7DE7D083BE156
A187CF4F0B18190952C2E3FBF8872815B09DD6635C0F53417494354BDD47F1F95A6
503BF237D430E39CE737F7D86340C96AEB9974E1C277EDBFE20C607BAF8C1020999
468E09B004979C3A0AEAE773E814646E73E33900D6C1D55FFF6DC40EF8D580617DA
C0D72DF0A9B7D5F87EEF5D02AD046991AB742D63C35B58CD59C3BED338555876242
4EE8BEDA972026AA1263CAFE1B41754F4D70F5E25F3
```

In [5]:
```
# Создам объект класса RSA для шифрования файлов и
# передам ему ключ для шифрования (открытый)

from rsa import RSA

encryptor = RSA(public_key_path="key.public")
encryptor
```

Out[5]:
```
<rsa.RSA at 0x7f9369fc2e90>
```

In [6]:
```
# Для наглядности вывожу в строчном представлении
# модели и её основные параметры. Видим что объект
# класса обладает только ключами для зашифрования

encryptor.__str__()
```

Out[6]:
{'public_key': '0XA7733914AF645573B787DDD55F1CA35CF3FC0DA25639A97F7
84A343D4223A4E99F784969A54D8D71FEA3BADB57B8C25ADC6F0367C30CA023E361
D35DF5DE1243CE6FB2FDC386D3B9F371D2F8B360584F7DABE642146A00C77E51E2B
F802199D3FF3C892D73A39B2F5C0BE9FB261C32D1E9F3186C98424B99748347FF0E
AA1217F3D5EC49D1EEB0CF0D9606309A1E778B2BAC6AD19E71D0A8B128C8D055C8F
BF30662DD0B41BAADFFF8C5CB2F352D3154FAEF5B424DA62440C7A6C7786CD6ACB3
24479288E8CDA01485AFCACE3DF0AD9824DBB6932CA32778A332BA09E5C88B9C621
CB4D1A663276371B3CE692AAE131729BE11BEA7BE545A8B0EB130CC966D3F',
 'public_key_int': 211386154944227170267489187876316867482387530591
157969469491411767877433568885413093407760707033488961653048748004 1
612304218234386387630558075618674653493055329054417047982977502683 6
572768509153574530859656854528610617212330097231292491356864845519 6
472454917921211144760580832383328654427094581676286101514398441416 1
144772335934315166849029998681578918881873745508444910492779875686 5
886713206978917513513577156729502250566865739291622126545508579744 3
282628658383966072046100060161349179103090533871176728969699748453 9
644313014593063111704452628920588441000121707795663714532092178745 3
793890838782501407513705394367807,
 'public_key_size': 2048,
 'secret_key': None,
 'secret_key_int': None,
 'secret_key_size': None,
 'module': '0X86C7A34204FEB43E5C3B4940AD51819F0F729F892EF3B0F727475
9D4C7BFB24B4940DC676DDEA556849E3260288454B197CF90CEBE2FF14FD926B3DB
72B5E374256294AE0E8DC2E015579DC152ED78AD54EC273194194812DF0368A2EE1
4C7FE9574452ACB23B477817BD961B47E7101ED529A4FB771F842E042DF4F799835
851B569F4FADC389940FEDBAAAF151213EAE9EC51E2E0F6428005125C4EE3D5B06B
64820752FA18DF042DB53963B85E8A8AC462F14778FB37A5DA2DA5CAAA4157CE23C
A44F3C52C50E21FC8737139C61609A0D2A47093AB29F7386A617DB7CD0B15C98862
8601B243876D39459092CCBEE843A67BBBB908935A97FEBEB50F033CAAE328D4C84
A6F187B45EE6DF7FD4FF0CDDBF8E6378AD308C4FD5F8C08648B77D75036CB9B034D
EF6CDCAB27691917465F509B49E591DC591AC7A86B78C5C6D4AEF3A222617AD670D
EA4EE22AD63FF41F948BB32B827D7ECE477F9866CD2E7DE7D083BE156A187CF4F0B
18190952C2E3FBF8872815B09DD6635C0F53417494354BDD47F1F95A6503BF237D4
30E39CE737F7D86340C96AEB9974E1C277EDBFE20C607BAF8C1020999468E09B004
979C3A0AEAE773E814646E73E33900D6C1D55FFF6DC40EF8D580617DAC0D72DF0A9
B7D5F87EEF5D02AD046991AB742D63C35B58CD59C3BED3385558762424EE8BEDA97
2026AA1263CAFE1B41754F4D70F5E25F3',
 'module_int': 549853753806690391887839250713982549030937317542518 1
774944152097950566364707022887603884178531900712473849398113381197 2
571661020469059335509817507217216391017923869449432796440654579732 4
105340797918945320410425405415008686196996176865393431734552716480 4
353093187753113255261930651625381565886538005929110251510658211390 9
182577793444378415361082623513241851336726274498272528374526656279 7
262984505669928439794765498490576729168937325374703806639360697298 3
448499074247061243625400727030172229482751108154337571483444205413 6
010783673181705378924202977915384541681390322489692015719843592060
290085925064034458025826970133082746480502416783656983333328894293
323034245676444072242777575984576790168994537967507136868290352181 8
247215891195901678003357841520335808386718218992928778865612182662 4
804130845839763894644231703044567404140599767422608409489082866526 9
635533871154951941202309884422864484136144219309591682370149884460 2
593123844871227548335590479055091509824590693873712472213114516688 6
759155926757109770837178369055775681460173859732591990642479169259 7
270554608630294721547560451779251470897915373347026270850735416677 0
148848913493662241036710003233738783245167115525953339766763140009 4
344423776236469168396028721893029428733427,
 'block_size': 4095,
 'block_size_full': 4096}

In [7]:
```
# Создам отдельный объект для расштфрования и
# передам ему закрытый ключ

decryptor = RSA(secret_key_path="key.secret")
decryptor
```

Out[7]:
```
<rsa.RSA at 0x7f9369651bd0>
```

In [8]:
```
# Также пристально посмотрим на объект через специальный метод
# Этот объект напротив - содержит лишь ключи для расшифрования
# Таким образом можно ограничить функциональность класса просто
# передав или не передавая ей нужные данные.

decryptor.__str__()
```

Out[8]:

```
{'public_key': None,
 'public_key_int': None,
 'public_key_size': None,
 'secret_key': '0X45470070F08F341787D931C8291287A4394B387086E8F29ED
105E76FB058F59B86109BDC3B25EB0396E8C5F055B7CB29E26074C99227113CFEBC
F6C009DEF7EED47E82EAB95C840EFAFA7017A745CF31E7C3B637F4D720694EA2A4A
52C345EAD99F6CC67BA0D1DAFF5FB86B02804D207AF26AA14960D3374C3F0938E63
0A193FBF4544CFDA9E94015FE35D4FDC6F964229EB7A0F79AD762A91F1899641C5B
A8036609B45A8A481D5F3DBEDAAB8972EB1A04345D0170803FC76C60AB2BC03C92A
9E41EB26A80915C167545570F0B2E2FEB4C3B324545E318F93D8B5E909B9E86FE14
BE16A918AFD7E3DDB0277D8BFDBE6EE8D51BB5C20CA837C9AA7EE13536841A57C85
8C699752E3EE65B39625D88D6075F98F947B0B7895B74B777BAA9D5C5407496E3ED
C84D201E8CBEE2F32B998DB78B1512FC42D9C3BE40CB29F0270E24617CEE20E3E04
210F47FBF2F2785B62D6037B39087AED14DE7ED322729545BA819C366F12F63B5D2
9415A86632D8416C445F3CD52F8D4095ADDA03B5D96886F0EE9FCD4B7F058027A34
38AA7A23894BCA743A625980F3128B1F7245C960EB70335FD4B2DDD1D7D9CE24B98
DB21E254563288258E8D0EB1F2B62E8D4B55BFF7AB68BA5C7426D6A54CC9E4B589D
3B1C54E36C26C384D3E5162C77279625F0608EBFC47FB8755DFBF2CDFC36A16D113
10CB049F87AE50DE01CCA7A128E36F93E6A5F',
 'secret_key_int': 282626931940940314234408806868721788655649415624
331743753929656547037022746890800369287973849873306308202663306273 6
065911590564761442313200000039547362764958281288908708873888969385 1
802633262478052914375124996175430660108982359109296515347151099775 1
951011537729391365607708545693887700472154810518460866072658387550 5
195441774351011183527814380364029459417814353687513508022507506381 5
359407282257475967515749845925215283976610580189279068463385647805 0
266159119344022727431400651648613066612833844345854526567710482499 5
704570858528550142325459802485202528424349611303688874542478536399 5
556471311358551377524412947871092988396741401142839357168605583442 7
628456762308865811538099116930370770139037759648926179650558708648 9
983757896717779900593911032298575280175882792957537457400727133337 7
954010076637595658653244402826626903177280654693691749262213915590 5
813557895209070257695306167248462904304795029065185538056857090803 7
981825164898801690457526088270037840162188071703654168671928577226 0
102449348641560050661495662658993508141145815456347389534092946155 1
073610562924451354660950627702944027004092607333684313573541907984 4
110227604515409525431697868492604409331492846319882088420913789303 6
667503906270900789658429462510114806295838166 3,
 'secret_key_size': 4095,
 'module': '0X86C7A34204FEB43E5C3B4940AD51819F0F729F892EF3B0F727475
9D4C7BFB24B4940DC676DDEA556849E3260288454B197CF90CEBE2FF14FD926B3DB
72B5E374256294AE0E8DC2E015579DC152ED78AD54EC273194194812DF0368A2EE1
4C7FE9574452ACB23B477817BD961B47E7101ED529A4FB771F842E042DF4F799835
851B569F4FADC389940FEDBAAAF151213EAE9EC51E2E0F6428005125C4EE3D5B06B
64820752FA18DF042DB53963B85E8A8AC462F14778FB37A5DA2DA5CAAA4157CE23C
A44F3C52C50E21FC8737139C61609A0D2A47093AB29F7386A617DB7CD0B15C98862
8601B243876D39459092CCBEE843A67BBBB908935A97FEBEB50F033CAAE328D4C84
A6F187B45EE6DF7FD4FF0CDDBF8E6378AD308C4FD5F8C08648B77D75036CB9B034D
EF6CDCAB27691917465F509B49E591DC591AC7A86B78C5C6D4AEF3A222617AD670D
EA4EE22AD63FF41F948BB32B827D7ECE477F9866CD2E7DE7D083BE156A187CF4F0B
18190952C2E3FBF8872815B09DD6635C0F53417494354BDD47F1F95A6503BF237D4
30E39CE737F7D86340C96AEB9974E1C277EDBFE20C607BAF8C1020999468E09B004
979C3A0AEAE773E814646E73E33900D6C1D55FFF6DC40EF8D580617DAC0D72DF0A9
B7D5F87EEF5D02AD046991AB742D63C35B58CD59C3BED3385558762424EE8BEDA97
2026AA1263CAFE1B41754F4D70F5E25F3',
 'module_int': 549853753806690391887839250713982549030937317542518 1
774944152097950566364707022887603884178531900712473849398113381197 2
571661020469059335509817507217216391017923869449432796440654579732 4
105340797918945320410425405415008686196996176865393431734552716480 4
353093187753113255261930651625381565886538005929110251510658211390 9
182577793444378415361082623513241851336726274498272528374526656279 7
```

```
26298450566992843979476549849057672916893732537470380663936069729834
44849907424706124362540072703017222948275110815433757148344420541360
10783673181705378924202977915384541681390332248969201571984359206029
00859250640344580258269701330827464805024167836569833333288942933
23034245676440722427775759845767901689945379675071368682903521818247
21589119590167800335784152033580838671821899292877886561218266248041
30845839763894644231703044567404140599767422608409489082866526963553
38711549519412023098844228644841361442193095916823701498844602593123
84487122754833559047905509150982459069387371247221311451668867591559
26757109770837178369055775681460173859732591990642479169259727055460
86302947215475604517792514708979153733470262708507354166770148848913
49366224103671000323373878324516711552595333976676314000943444237762
36469168396028721893029428733427,
 'block_size': 4095,
 'block_size_full': 4096}
```

In [9]:
```
# Хотя можно было все сделать одним объектов
# просто передав ему оба ключа

rsa = RSA(public_key_path="key.public", secret_key_path="key.secret"
rsa.__str__()
```

Out[9]:
{'public_key': '0XA7733914AF645573B787DDD55F1CA35CF3FC0DA25639A97F7
84A343D4223A4E99F784969A54D8D71FEA3BADB57B8C25ADC6F0367C30CA023E361
D35DF5DE1243CE6FB2FDC386D3B9F371D2F8B360584F7DABE642146A00C77E51E2B
F802199D3FF3C892D73A39B2F5C0BE9FB261C32D1E9F3186C98424B99748347FF0E
AA1217F3D5EC49D1EEB0CF0D9606309A1E778B2BAC6AD19E71D0A8B128C8D055C8F
BF30662DD0B41BAADFFF8C5CB2F352D3154FAEF5B424DA62440C7A6C7786CD6ACB3
24479288E8CDA01485AFCACE3DF0AD9824DBB6932CA32778A332BA09E5C88B9C621
CB4D1A663276371B3CE692AAE131729BE11BEA7BE545A8B0EB130CC966D3F',
 'public_key_int': 211386154944227170267489187876316867482387530591
157969469491411767877433568885413093407760707033488961653048748004 1
612304218234386387630558075618674653493055329054417047982977502683 6
572768509153574530859656854528610617212330097231292491356864845519 6
472454917921211144760580832383328654427094581676286101514398441416 1
144772335934315166849029998681578918881873745508444910492779875686 5
886713206978917513513577156729502250566865739291622126545508579744 3
282628658383966072046100060161349179103090533871176728969699748453 9
644313014593063111704452628920588441000121707795663714532092178745 3
793890838782501407513705394367807,
 'public_key_size': 2048,
 'secret_key': '0X45470070F08F341787D931C8291287A4394B387086E8F29ED
105E76FB058F59B86109BDC3B25EB0396E8C5F055B7CB29E26074C99227113CFEBC
F6C009DEF7EED47E82EAB95C840EFAFA7017A745CF31E7C3B637F4D720694EA2A4A
52C345EAD99F6CC67BA0D1DAFF5FB86B02804D207AF26AA14960D3374C3F0938E63
0A193FBF4544CFDA9E94015FE35D4FDC6F964229EB7A0F79AD762A91F1899641C5B
A8036609B45A8A481D5F3DBEDAAB8972EB1A04345D0170803FC76C60AB2BC03C92A
9E41EB26A80915C167545570F0B2E2FEB4C3B324545E318F93D8B5E909B9E86FE14
BE16A918AFD7E3DDB0277D8BFDBE6EE8D51BB5C20CA837C9AA7EE13536841A57C85
8C699752E3EE65B39625D88D6075F98F947B0B7895B74B777BAA9D5C5407496E3ED
C84D201E8CBEE2F32B998DB78B1512FC42D9C3BE40CB29F0270E24617CEE20E3E04
210F47FBF2F2785B62D6037B39087AED14DE7ED322729545BA819C366F12F63B5D2
9415A86632D8416C445F3CD52F8D4095ADDA03B5D96886F0EE9FCD4B7F058027A34
38AA7A23894BCA743A625980F3128B1F7245C960EB70335FD4B2DDD1D7D9CE24B98
DB21E254563288258E8D0EB1F2B62E8D4B55BFF7AB68BA5C7426D6A54CC9E4B589D
3B1C54E36C26C384D3E5162C77279625F0608EBFC47FB8755DFBF2CDFC36A16D113
10CB049F87AE50DE01CCA7A128E36F93E6A5F',
 'secret_key_int': 282626931940940314234408806868721788655649415624
331743753929656547037022746890800369287973849873306308202663306273 6
065911590564761442313200000039547362764958281288908708873888969385 1
802633262478052914375124996175430660108982359109296515347151099775 1
951011537729391365607708545693887700472154810518460866072658387550 5
195441774351011183527814380364029459417814353687513508022507506381 5
359407282257475967515749845925215283976610580189279068463385647805 0
266159119344022727431400651648613066612833844345854526567710482499 5
704570858528550142325459802485202528424349611303688874542478536399 5
556471311358551377524412947871092988396741401142839357168605583442 7
628456762308865811538099116930370770139037759648926179650558708648 9
983757896717779900593911032298575280175882792957537457400727133337 7
954010076637595658653244402826626903177280654693691749262213915590 5
813557895209070257695306167248462904304795029065185538056857090803 7
981825164898801690457526088270037840162188071703654168671928577226 0
102449348641560050661495662658993508141145815456347389534092946155 1
073610562924451354660950627702944027004092607333684313573541907984 4
110227604515409525431697868492604409331492846319882088420913789303 6
667503906270900789658429462510114806295838166 3,
 'secret_key_size': 4095,
 'module': '0X86C7A34204FEB43E5C3B4940AD51819F0F729F892EF3B0F727475
9D4C7BFB24B4940DC676DDEA556849E3260288454B197CF90CEBE2FF14FD926B3DB
72B5E374256294AE0E8DC2E015579DC152ED78AD54EC273194194812DF0368A2EE1
4C7FE9574452ACB23B477817BD961B47E7101ED529A4FB771F842E042DF4F799835
851B569F4FADC389940FEDBAAAF151213EAE9EC51E2E0F6428005125C4EE3D5B06B
64820752FA18DF042DB53963B85E8A8AC462F14778FB37A5DA2DA5CAAA4157CE23C

```
A44F3C52C50E21FC8737139C61609A0D2A47093AB29F7386A617DB7CD0B15C98862
8601B243876D39459092CCBEE843A67BBBB908935A97FEBEB50F033CAAE328D4C84
A6F187B45EE6DF7FD4FF0CDDBF8E6378AD308C4FD5F8C08648B77D75036CB9B034D
EF6CDCAB27691917465F509B49E591DC591AC7A86B78C5C6D4AEF3A222617AD670D
EA4EE22AD63FF41F948BB32B827D7ECE477F9866CD2E7DE7D083BE156A187CF4F0B
18190952C2E3FBF8872815B09DD6635C0F53417494354BDD47F1F95A6503BF237D4
30E39CE737F7D86340C96AEB9974E1C277EDBFE20C607BAF8C1020999468E09B004
979C3A0AEAE773E814646E73E33900D6C1D55FFF6DC40EF8D580617DAC0D72DF0A9
B7D5F87EEF5D02AD046991AB742D63C35B58CD59C3BED3385558762424EE8BEDA97
2026AA1263CAFE1B41754F4D70F5E25F3',
 'module_int': 549853753806690391887839250713982549030937317542518177
749441520979505663647070228876038841785319007124738493981133811972
571661020469059335509817507217216391017923869449432796440654579732410
534079791894532041042540541500868619699617686539343173455271648043530
931877531132552619306516253815658865380059291102515106582113909182577
793444378415361082623513241851336726274498272528374526656279726298
450566992843979476549849057672916893732537470380663936069729834484
990742470612436254007270301722294827511081543375714834442054136010
783673181705378924202977915384541681390332248969201571984359206029008
592506403445802582697013308274648050241678365698333333288942933230
342456764440722427775759845767901689945379675071368682903521818247
215891195901678003357841520335808386718218992928778865612182662480
413084583976389464423170304456740414059976742260840948908286652696
355338711549519412023098844228644841361442193095916823701498844602
593123844871227548335590479055091509824590693873712472213114516688675
915592675710977083717836905577568146017385973259199064247916925972
705546086302947215475604517792514708979153733470262708507354166770
148848913493662241036710003233738783245167115525953339766763140009434
442377623646916839602872189302942873342 7,
 'block_size': 4095,
 'block_size_full': 4096}
```

In [10]:
```
# Проверим правильность работы ключей у объектов encrypted
# и decrypted. Сгенерируем случайное число и зашифруем а
# затем расшифруем их

import random

_int = random.randrange(2, 100)

encr = pow(_int, encryptor.public_key_int, encryptor.module_int)
decr = pow(encr, decryptor.secret_key_int, decryptor.module_int)

bool(decr == _int)
```

Out[10]: True

In [11]:
```
# Проделаем то же самое с объектом rsa чтобы подтвердить
# корректность суждений

_int = random.randrange(2, 100)

encr = pow(_int, rsa.public_key_int, rsa.module_int)
decr = pow(encr, rsa.secret_key_int, rsa.module_int)

bool(decr == _int)
```

Out[11]: True

Часть 2. Зашифрование и расширование

In [12]:
```
# Для демонстрации зашифрования и расшифрования мною были
# подготовлены два файла с тескстом open.txt и изображением
# image.png

# Передам методу encrypt относительный путь к файлу для
# зашифрования и название файлы куда дб записан зашифрованый
# файл encrypted.txt

rsa.encrypt("open.txt", "encrypted.txt")
```

In [13]:
```
# Попробуем прочитать зашифрованый файл

!cat encrypted.txt
```

```
kV������e◄◅◁��p������ᛉ@i,g7J,%#K��-��~@@��ᛊ0
����LIG⍉dpp>'оԋqB&GY№=g☎E▷I♪Z№U⍉I
gS№I⍉ᛊ⍉ek_n⌕Y>Np⍉ز⍉,⍉ᛉI⍉),@`L№g⌕3◷⍉♀
⍉Y?⍉⍉m1⍉⍉p⍉⍉~№UZ ˬ}⍉ş⍉ ⍉⍉⍉¨⍉y⌀4
⍉9♪n⍉%⍉⍉~⍉⍉$¨⍉p⍉v⍉[⍉N⍉O⍉#4⍉f⍉E#⍉⍉⍉¤⍉耻uk◄⍉⍉q⍉
⍉⍉⍉ᛊʸK⍉⍉~⍉⍉⍉⍉⍉⍉⍉F}nO⍉⍉$⍉p
T⌕⍉⍉gx⍉⍉/K⍉⍉z,⍉ᛊe_⍉8⍉ȥc⍉kز⍉⍉Pe⍉@⍉⍉⍉⍉0⍉z ⍉⍉4:⍉-Y⍉
P。ӝ⍉⍉W⍉-J∝⍉80|⍉(⌐v⍉k⍉G^Q^⍉⍉PƑ⍉hd⍉⍉⍉fV⍉îN⍉⍉o⍉C⍉⍉ě
⍉⍉
Ƀ⍉⍉V(⍉~o⍉⍉A⍉7⍉⍉<⍉⍉s⍉⍉☎⍉⍉⍉⍉⍉B⍉⍉⍉⍉∝⍉⍉⍉⍉⍉⍉
⍉/⍉⍉r⍉X♫x⍉w⍉⍉{&⍉⍉\⍉⍉+⍉.z26⍉⍉♀/E⍉⍉8
```

In [14]:
```
# Передам методу decrypt относительный путь к файлу для
# расшифрования и название файлы куда дб записан расшифрованый
# файл decrypted.txt

rsa.decrypt("encrypted.txt", "decrypted.txt")
```

In [15]:
```
!cat decrypted.txt
```

```
Прощай немытая Россия,
Страна рабов, страна господ,
И вы, мундиры голубые,
И ты, им преданный народ.

Быть может, за стеной Кавказа
Сокроюсь от твоих пашей,
От их всевидящего глаза,
От их всеслышащих ушей.

Михаил Лермонтов, 1841г.
```

In [16]:
```
# Проверим корректность расшифровки

with open("open.txt", "rb") as file:
    open_text = file.read()

with open("decrypted.txt", "rb") as file:
    decrypted_text = file.read()

bool(open_text == decrypted_text)
```

Out[16]:
```
True
```

In [17]:
```
# Установлю библиотеку для просмотра изображений

!pip3 install pillow
```

```
Defaulting to user installation because normal site-packages is not
writeable
Requirement already satisfied: pillow in /usr/lib/python3/dist-pack
ages (9.4.0)
```

In [18]:
```
# Зашифрую изображение и сохраню в файл encrypted.png

rsa.encrypt("image.png", "encrypted.png")
!ls -l
```

```
total 1352
-rw-r--r-- 1 evgeny evgeny 431919 Apr 25 19:45 analyze.ipynb
-rw-r--r-- 1 evgeny evgeny    408 Apr 29 10:29 decrypted.txt
-rw-r--r-- 1 evgeny evgeny  88771 Apr 29 10:29 demo.ipynb
-rw-r--r-- 1 evgeny evgeny  35328 Apr 29 10:30 encrypted.png
-rw-r--r-- 1 evgeny evgeny    512 Apr 29 10:29 encrypted.txt
-rw-r--r-- 1 evgeny evgeny  35117 Apr 23 16:57 image.png
-rw-r--r-- 1 evgeny evgeny   4172 Apr 26 17:00 keygen.py
-rw-r--r-- 1 evgeny evgeny   1541 Apr 29 10:28 key.public
-rw-r--r-- 1 evgeny evgeny   2053 Apr 29 10:28 key.secret
-rw-r--r-- 1 evgeny evgeny  24721 Apr 29 10:25 manual.ipynb
-rw-r--r-- 1 evgeny evgeny 399380 Apr 29 10:27 manual.pdf
-rw-r--r-- 1 evgeny evgeny 306562 Apr 25 16:42 mayday.png
-rw-r--r-- 1 evgeny evgeny    408 Apr 20 15:36 open.txt
drwxr-xr-x 2 evgeny evgeny   4096 Apr 26 17:09 __pycache__
-rw-r--r-- 1 evgeny evgeny   5100 Apr 25 19:02 rsa.py
-rw-r--r-- 1 evgeny evgeny     13 Apr 25 17:06 secret.crack
drwxr-xr-x 6 evgeny evgeny   4096 Apr 15 15:18 venv
```

In [19]:
```
# Попробую открыть файл

from PIL import Image

try:
    Image.open("encrypted.png")
except:
    print("Невозможно прочитать файл!")
```

```
Невозможно прочитать файл!
```

In [20]:
```
# Расшифрую изображение и сохраню в файл decrypted.png

rsa.decrypt("encrypted.png", "decrypted.png")
!ls -l
```

```
total 1388
-rw-r--r-- 1 evgeny evgeny 431919 Apr 25 19:45 analyze.ipynb
-rw-r--r-- 1 evgeny evgeny  35117 Apr 29 10:30 decrypted.png
-rw-r--r-- 1 evgeny evgeny    408 Apr 29 10:29 decrypted.txt
-rw-r--r-- 1 evgeny evgeny  88771 Apr 29 10:29 demo.ipynb
-rw-r--r-- 1 evgeny evgeny  35328 Apr 29 10:30 encrypted.png
-rw-r--r-- 1 evgeny evgeny    512 Apr 29 10:29 encrypted.txt
-rw-r--r-- 1 evgeny evgeny  35117 Apr 23 16:57 image.png
-rw-r--r-- 1 evgeny evgeny   4172 Apr 26 17:00 keygen.py
-rw-r--r-- 1 evgeny evgeny   1541 Apr 29 10:28 key.public
-rw-r--r-- 1 evgeny evgeny   2053 Apr 29 10:28 key.secret
-rw-r--r-- 1 evgeny evgeny  24721 Apr 29 10:25 manual.ipynb
-rw-r--r-- 1 evgeny evgeny 399380 Apr 29 10:27 manual.pdf
-rw-r--r-- 1 evgeny evgeny 306562 Apr 25 16:42 mayday.png
-rw-r--r-- 1 evgeny evgeny    408 Apr 20 15:36 open.txt
drwxr-xr-x 2 evgeny evgeny   4096 Apr 26 17:09 __pycache__
-rw-r--r-- 1 evgeny evgeny   5100 Apr 25 19:02 rsa.py
-rw-r--r-- 1 evgeny evgeny     13 Apr 25 17:06 secret.crack
drwxr-xr-x 6 evgeny evgeny   4096 Apr 15 15:18 venv
```

In [21]:
```
# Проверим корректность расшифровки

with open("image.png", "rb") as file:
    source_image = file.read()

with open("decrypted.png", "rb") as file:
    decrypted_image = file.read()

bool(source_image == decrypted_image)
```

Out[21]: True

In [22]:
```
# Попробую открыть расшифрованый файл

Image.open("decrypted.png")
```

Out[22]: