

КРИПТОГРАФІЯ КОМП'ЮТЕРНИЙ

ПРАКТИКУМ №4

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

Виконали: Медвецький Давид та Левашова
Світлана

Група: ФБ-13

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 зашкереженого зв'язку й електронного підпису, вивчення протоколу розсилання ключів.

Хід роботи:

```
Абонент А:  
p_A = 67434069773723700423316764014229821814315609807542663121007590065672672455501  
q_A = 69300489368712328126289055783103480381950136620345764968554336794560106630243  
Абонент В:  
p_B = 83096649198860189408834193720986319302458639378116757478126020169608016661751  
q_B = 107937475688991463934544371865817480325178374657786580984341088557059596106573
```

- наші випадкові прості числа.

```
Все кандидаты:  
95775379046696177425064814988105572238579338090656456194120350398957312369936  
89241919288289610751564424841433197125847655292969133170638472221894832589624  
92781241488280728534723841546759559951528909715769063030494480432022538707842  
103101469730663788682235382656807068122737094793328082780489701014829465617797  
86887175407066854492714746637164057309787602169012296874323958395026740926079  
79735559463503153938303599565848599014339976448218159078553709807916245791109  
74544696066954301934628943197496084610205103880773697157289371933735420418913  
77936258097381253422023087547085419458826831806695076753024715745149311715608  
112952982437642828993645871248737369454728290822508847741768323038143377186520  
75887821412746462127561706928338401560408198302384593024634574060668918210596  
108046685252286868180304900698493237283339299609779585520243011687714192526730  
64099711424831102124566701319958406217375711774830858013830379355264254237715  
86044210984897342866324109387746465115814040842540194237464742279832841346715  
70686080599889273465399370625093905155879834331131095416251678505935869536493  
59617579511104163767477609909076359931972283891641607026107157970289326234492  
115364010255583583679136724238814877579859699950381425044119666420173840264160  
77312212619181996076008835523353354736071559067409354196054509465670378946712
```

- всі кандидати (виводили суто для протоколу)

```
Ключи  
Открытый ключ А (n, e): (46732140354429446515256425592749238449143103041090126549898186372144735547694396676223004  
Секретный ключ А (n, d): (46732140354429446515256425592749238449143103041090126549898186372144735547694396676223004  
Открытый ключ В (n, e): (89692425527386237023892712268568751321411195721400509300220856196756115307056602211929168  
Секретный ключ В (n, d): (89692425527386237023892712268568751321411195721400509300220856196756115307056602211929168
```

- параметри криптосистеми RSA для абонентів А та В (так як числа занадто великі, на скріні не видно відкриті та секретні ключі повністю).

```
Абонент А:  
Сообщение: 36145440483680697188271279333578939792379624557734109568456938326412176240950  
Зашифрованное сообщение: 467827787360616382308175221901763727194312057659286483864462854972159504079054097  
Подпись: 1745131766268541978982081517189550008971223380252667611066078327502375800418604360029566843566019  
  
Абонент В:  
Разшифрованное сообщение: 36145440483680697188271279333578939792379624557734109568456938326412176240950  
Проверка цифровой подписи: True
```

- згенероване відкрите повідомлення, зашифроване повідомлення, цифровий підпис та перевірка цифрового підпису.

Як в цілому працює код:

Спочатку для абонентів А і В генеруються два простих числа p і q .

Наступним кроком для абонентів А і В генеруються відкриті та секретні ключі з використанням простих чисел, що були згенеровані у попередньому кроці (p , q).

Далі генерується випадкове повідомлення для абонента А. Це повідомлення шифрується відкритим ключем абонента В. Отримане зашифроване повідомлення розшифровується секретним ключем абонента В.

Перевірка підпису:

Береться випадкове повідомлення для абонента А, що підписується секретним ключем абонента А. І підпис перевіряється відкритим ключем абонента А.

Перевірка операцій на [тестовому середовищі](#):

```
8969242552738623702389271226856875132141119572140050930022085619675611530705660221192916878343502615602920812045615024291003583416124912579915872288789323
```

- наш модуль з публічного ключа В.

```
6808506458901645578500462902808890529067784282217005072695199512549176758043665582195672954060925517421497051228677294928618971758587354314808218967095837
```

- публічна експонента.

```
36145440483680697188271279333578939792379624557734109568456938326412176240950
```

- згенероване повідомлення.

Encryption

✖ Clear

Modulus

AB40BE8EA12CA351F01E87EBB23460FEB85DF6BEC95B483C1F4D66F82A523A58315C82924E548B315EF1E

Public exponent

81FF494D91E5A7C78073B2D755E21666805D434A8170BB6C133F601274506D232F11DC62E234D0093536E9E

Message

4FE998259D1E6F69600544274F6ECAD441F81B7D44C9F471A75ABD7A9B41B1

Bytes ▼

Encrypt

Ciphertext

5952F22740252A1B9FC4E10C4AE26ED625F83437D339648AEDA2C8ACE2F57DFBB45DB6854A33654F32768

- перевели в шістнадцяткову систему числення та отримали шифртекст.

Hexadecimal to Decimal converter

From

Hexadecimal ▼

To

Decimal ▼

Enter hex number

5952F22740252A1B9FC4E10C4AE26ED625F

16

= Convert

✖ Reset

↕ Swap

Decimal number (154 digits)


467827787360616382308175221901763
727194312057659286483864462854972

10

перевели в десяткову систему числення та отримали правильне розшифрування.

Зашифрованное сообщение: 4678277873606163823081752219017637271943120576592864838644628549721595040790540973387633899531568356005008976693862684

Get server key



Key size

256

Get key

Modulus

9378B744CA24C37EEF2FC1BF3A00A26E3DA80927B7E0B2D8BB57755155E65BA5

Public exponent

10001

- згенерували модуль та експоненту.


```
# Шифруємо і розшифровуємо повідомлення для абонента А
message_A = 36145440483680697188271279333578939792379624557734109568456938326412176240950
encrypted_message_A = encrypt(message_A, n: 66703275263659607857607982718184123795922479827902579188919434789179921685413, e: 65537)
decrypted_message_A = decrypt(encrypted_message_A, private_key_B)
```

- шифруємо нашою програмою.

Сообщение: 36145440483680697188271279333578939792379624557734109568456938326412176240950

- отримуємо повідомлення.

Decryption



Ciphertext

502394D9C13C31A705FA66ABC37D2C135FAC28041E0D6A7E643833C7BE135C

Bytes ▼

Decrypt

Message

4FE998259D1E6F69600544274F6ECAD441F81B7D44C9F471A75ABD7A9B41B136

- переводимо наше повідомлення в hex та дешифуємо.

Hexadecimal to Decimal converter

From

To

Hexadecimal

Decimal

Enter hex number

4FE998259D1E6F69600544274F6ECAD441F

16

= Convert

× Reset

↕ Swap

Decimal number (77 digits)

361454404836806971882712793335789
397923796245577341095684569383264

10

- бачимо, що все співпадає.

```
lab2_cp.py 153 # Шифруєм і розшифровуємо повідомлення для абонента А
lab2_cp.py 154 message_A = 36145440483680697188271279333578939792379624557734109568456938326412176240950
lab3_cp.py 155 encrypted_message_A = encrypt(message_A, public_key_B)
main.py 156 decrypted_message_A = decrypt(encrypted_message_A, private_key_B)
mamed.py 157
test.py 158 # Абонент В перевіряє підпис абонента А
побудов 159 signature_A = 15322894971864458837111217670799079568723211943626639538529031185796262247336
External Lib 160 verification_result_A = verify(signature_A, message_A, in: 66703275263659607857607982718184123795922479827902579188919434789179921685413, e: 65537)
Scratches a 161 print("\nАбонент А:")
162 print("Сообщение:", message_A)
```

CP4_Medvetskiy_Levashova_FB-13 bla ×

Абонент А:
Сообщение: 36145440483680697188271279333578939792379624557734109568456938326412176240950
Зашифрованное сообщение: 33205496547620272797743142932885606509872402007233099521544378598385338622444269863874228599686800159681210769296602417477306699334221735030
Подпись: 15322894971864458837111217670799079568723211943626639538529031185796262247336

Абонент В:
Разшифрованное сообщение: 36145440483680697188271279333578939792379624557734109568456938326412176240950
Проверка цифровой подписи: True

- за допомогою згенерованих на сервері ключа та підпису проводимо перевірку та бачимо, що вона пройшла успішно.

Verify

 Clear

Message

4FE998259D1E6F69600544274F6ECAD441F81B7D44C9F471A75ABD7A9B41B1

Bytes ▾

Signature

215206C14C51D90A857D00F1BC5107FB28DC7F46392EE739A9D87F0C4847CB849520B5C3270E12C6B66CE

Modulus

593A31C349052CEB9B155092DB4BA92EF1BFD397C302D0835AACD91CAB4FF19F629CE975DC40777C4DB6

Public exponent

4B66F40AC3FA88FE5BC5A54524FB16A53E1429B05FD8B4BAF488EACD6A62106ECB37AACCD6C8B9D1654

Verify

Verification

true

✓

- взяли наше повідомлення, згенерований нами підпис, наше значення публічного ключа для абонента А і бачимо, що верифікація пройшла успішно.

Висновки:

Робота дозволила освоїти криптосистему RSA та алгоритм електронного підпису. Досліджено методи генерації ключів, перевірку чисел на простоту та протокол розсилання ключів. Практично використано систему RSA для засекреченого зв'язку та електронного підпису, розвиваючи навички в області криптографії.