



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 гр. ФБ-24 Тішевська Анна

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

Мета та основні завдання 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 і рекомендації щодо 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 $1 < p, q$ довжини щонайменше 256 біт. При цьому пари чисел беруться так, щоб $pq \leq p_1q_1$; p і q – прості числа для побудови ключів абонента А, $1 < p < q_1$ – 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, n) 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()`.

Результати роботи:

Програма виконала генерацію двох пар ключів RSA (для абонента А та абонента В), використовуючи прості числа p і q довжиною щонайменше 256 біт¹.

```
► Ключі абонента А
Відкритий ключ А (e, n):
eA = 42657985623929637616433771105731381578384976544329128680838018342408029687573243279274837028359025844061683286623229183596649853297525155341709632895997
nA = 46960982220239782671462019138384998614510703573419878196632346524679930999226586123343084031130700627714283203611490149751596492565529932523568937359503
Секретний ключ А (d, p, q):
dA = 27677920391029440791598162862887551866019218104002746239899486708239868997002968431560602291641768798907067891963015134189086319778621299302329735351349
pA = 92269654217771441203506042329738804910806916659211938725437882583243271947937
qA = 50895370334220827563488699041847176786678870975004138528988021572980724515119

► Ключі абонента В
Відкритий ключ В (e, n):
eB = 462388127236103696582109941615534029451076656627049870559062397479966007126920069573168999459330630056855687636217278790029893103575487633491736430997919
nB = 5368742402736491696672948360263853752548207304525600432713237247824589580168870159501189057786402850137925710696575371933740936384528517631806801975058889
Секретний ключ В (d, p, q):
dB = 2268590336673390026732395985358118824914417606222825511777658344332401979631289827134802111802933904685460365245600714279560900832382853552927012671746399
pB = 83775520431626897817359219835187267974966222093764025684477247890823906019929
qB = 64084858859434628040648037089285451028258844430281362187686883752984347264241

► Кандидати в прості, що не пройшли тест Міллера-Рабіна
Кількість: 374
Перш 10: 192269654217771441203506042329738804910806916659211938725437882583243271947763, 92269654217771441203506042329738804910806916659211938725437882583243271947765, 92269654217771441203506042329738804910806916659211938725437882583243271947767, 92269654217771441203506042329738804910806916659211938725437882583243271947769, 92269654217771441203506042329738804910806916659211938725437882583243271947771, 92269654217771441203506042329738804910806916659211938725437882583243271947773, 92269654217771441203506042329738804910806916659211938725437882583243271947775, 92269654217771441203506042329738804910806916659211938725437882583243271947777, 92269654217771441203506042329738804910806916659211938725437882583243271947779, 92269654217771441203506042329738804910806916659211938725437882583243271947781
```

Кількість кандидатів у прості числа, які не пройшли тест Міллера–Рабіна: 374. Це свідчить про те, що генератору довелося виконати 374 ітерації перевірки, перш ніж знайти чотири прості числа.

```
Вихідний текст M: 4561119
Шифртекст C = M^e mod n: 4174176971160563612916060187361556803151390391705415193771198456621011056808301614422639982232956269325951981775735847346554393933091073157184894360691603
Розшифрований текст M' = 4561119

► Цифровий підпис А
Підпис S = M'^d mod n: 4532457051847159439570607396626722019142804843252648653922264559240650519590846392853953822281468473107774143048911535212271253704262131771156689849349
Перевірка підпису verify(M', S, Pub_A): True
```

```
Сеансовий ключ k: 1438257447148738981581662143802368863343853685336891388829478112944815037837769063927303188361648632583498430308583798366271782855304325139520479927227353
```

```
ВІДПРАВКА КЛЮЧА (A - B)

► Дані, які надіслав А
k1 (k, зашифрований ключ В): 501095285048504907649676835827395517888474204291556338957750369800005236256805819262793622100789009989379804126660144315566316102194920378898628628798947
S (підпис k приватним ключем А): 34874445451448700390904825924749496811622706320854956683866639421676828231160615752019990306163201248070248005927953069410935516462435743929014181068870
S1 (S, зашифрований ключем В): 256320614923611576657407915481187783760933265153573865274376427287236008592912478207936095334089007997158029807699825456595230251587341231303024941907212
Пара (k1, S1), що надіслано Б по каналу: (501095285048504907649676835827395517888474204291556338957750369800005236256805819262793622100789009989379804126660144315566316102194920378898628628798947, 256320614923611576657407915481187783760933265153573865274376427287236008592912478207936095334089007997158029807699825456595230251587341231303024941907212)

ОТРИМАННЯ КЛЮЧА (B)

► Результат обробки повідомлення Б
k (розшифрований ключ): 1438257447148738981581662143802368863343853685336891388829478112944815037837769063927303188361648632583498430308583798366271782855304325139520479927227353
S (розшифрований підпис): 34874445451448700390904825924749496811622706320854956683866639421676828231160615752019990306163201248070248005927953069410935516462435743929014181068870
Перевірка підпису: OK
Отриманий ключ та результат перевірки: (1438257447148738981581662143802368863343853685336891388829478112944815037837769063927303188361648632583498430308583798366271782855304325139520479927227353, True)
```

Для процедури розкриття ключів спочатку генерується значення k , де $0 < k < n$. Далі формується повідомлення у вигляді пари (k_1, S_1) , де $k_1 = k^e \bmod n$, $S_1 = S^e \bmod n$, а $S = k^d \bmod n$.

Після цього абонент В отримує повідомлення і виконує такі кроки:

$$k = k_1^d \bmod n,$$

$$S = S_1^d \bmod n,$$

після чого перевіряє підпис за умовою $k = S^e \bmod n$.

Результат: перевіряється коректність підпису та відновлюється вихідне значення ключа.

Перевірка

RSA Testing Environment

Server Key

Encryption

Decryption

Signature

Verification

Send Key

Receive Key

Get server key

Clear

Key size256Get key

Modulus90829D6CAF6C2D76F691E6DC30FC86B8BAD615F148A6C6EF202573A70005FB5D

Public exponent10001

```
► Приклад 2: використання server key (256 біт)
Server key size: 256 bit
Server modulus (hex): 0x90829D6CAF6C2D76F691E6DC30FC86B8BAD615F148A6C6EF202573A70005FB5D
Server public exponent (hex): 0x10001
C = M^e mod server_m: 0x74e3b7cda52c6b597a8d637b5d1a969368ce1173c1737183c8a1c4366de55143
```

RSA Testing Environment

Server Key

Encryption

Decryption

Signature

Verification

Send Key

Receive Key

Encryption

Clear

Modulus654efea76464a587b4eaa413301b5359f7c205f0a6b4f47b0c6fbc051a1f4df9a37ab47aa63c59c96ded00c12110de9a

Public exponent4ac7dfdf0e05c7a7eb366084c8f450ebeb977693c5ed13b99b32c3bff89219bf8df226a89c5f6fdb15925d585bdba837f

Message4598dfBytes

Encrypt

Ciphertext33B28F9E0D9275F6796F7C0E17A7D8F88FC956873CDA96CB75EE119A19F2B058420BAD5F0E393611480C9f

```
► Приклад 1: шифрування для великого модуля m
M (десятькові): 4561119
M (hex): 0x4598df
m: 0x654efea76464a587b4aaa413301b5359f7c205f0a6b4f47b0c6fbc051a1f4df9a37ab47aa63c59c96ded00c12110de947d2f4f348ac56426749440f6165a77a3
e: 0x4ac7dfdf0e05c7a7eb366084c8f450eb977693c5ed13b99b32c3bff89219bf8df226a89c5f6fdb15025d585bdba8378c3befac3fbb7e85637b1fba04ac4efd
C = M^e mod m: 0x33b2f9e0d9275f6796f7c0e17a7d8f88fc956873cda96cb75ee119a19f2b058420bad5f0e393611480c90a8f34407058617bc28a401677364c9b0d78158157f
```

RSA Testing Environment

Server Key

Encryption

Decryption

Signature

Verification

Send Key

Receive Key

Decryption

Clear

Ciphertext74e3b7oda52c6b597a8d637b5d1a969368ce1173c1737183c8a1c4366de55143Bytes

Decrypt

Message4598DF

RSA Testing Environment

Server Key

Encryption

Decryption

Signature

Verification

Send Key

Receive Key

Sign

Clear

Message738c4410Bytes

Sign

Signature214C782342E3CA571B39FEC8DE69C686BEB2CF80638F8ED1A123C98061BE41D8

```
► Приклад 4: підпис із заданими ключами
M: 0x15b4324fee
m публічного ключа: 0x162f70735dac7292d654fbadc3d066d9879391be0ef02c9ca9182889a3424c118d8f56139bf3b5dcd6ad5ef2371e94c3b30dd0c22676c62d30f99fb082cb0365
e публічного ключа: 0x127f18e5e7f13fea8c6b2d5cd285eb9e650b473fec281720ee83dd2a4f4464cf8d2ce059c028a1ab0746af9c8bb9a50128a32af4c6cbd194250cee89233106ad
Підпис S = M^d mod (p*q): 0xc489ffd43243ca2d78782c7bfff399661897d7099e8687c0c46dc33f0558ca1fa956a57c3b23feeadb587a30986293fb38b829c12c5b4b713fed5960ab29c84c
```

RSA Testing Environment

[Server Key](#)
[Encryption](#)
[Decryption](#)
[Signature](#)
[Verification](#)
[Send Key](#)
[Receive Key](#)

Verify

✕ Clear

Message

15b4324fee

Bytes ▾

Signature

c489ffd43243ca2d78782c7bff399661897d7099e8687c0c46dc33f0558ca1fa956a57c3b23feeadb587a30986293fb3

Modulus

162f70735dac7292d654fbadc3d066d9879391be0ef02c9ca9182889a3424c118d8f56139bf3b5dcd6ad5ef2371e94c

Public exponent

127f18e5e7f13fea8c6b2d5cd285ebee650b473fec281720ee83dd2a4f4464cf8d2ce059c028a1ab0746af9c8bb9a50

Verify

Verification

true ✓

Висновок за результатами виконання роботи

Виконана робота являє собою повну практичну реалізацію криптосистеми RSA та демонстрацію її використання для засекреченого зв'язку (шифрування/розшифрування), електронного підпису та реалізації протоколу конфіденційного розсилання ключів з підтвердженням справжності.