

Міністерство освіти і науки України
Національний технічний університет України
"Київський політехнічний інститут імені Ігоря Сікорського"
Фізико-технічний інститут

КРИПТОГРАФІЯ
КОМП'ЮТЕРНИЙ ПРАКТИКУМ №4

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

Виконали:
Студенти ФБ-01
Сотнікова П.О.
Струкало В.В.

Київ – 2022

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 p_1, q_1 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 A , p_1 і q_1 – 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) , (e_1, n_1) та секретні d і d_1 .
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 і 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.

*******Ключі для абонента А*******

p: 1109056885871486409645391802447405809967805671564405217089856497961759
34165547
q: 7322676473375683627772062163699423732620094408076104511892677376620918
3271219
e: 39439018651553844785137179456131351294516522152470544560398013533279163
16445111736219259495667697718928859608887107286241395236596171220366363718
869104287
n: 81212647658064341477393367538771555649744466981988938920288855262216439
78300164782927366968103560386072824276930351527334390653625410483145594306
546491793
d: 60576924343923152353415431233320534584387209189594166273196050000877459
45399905158537895853030916102005174114360222401628445047461978975599298074
487071199

*******Ключі для абонента В*******

p1: 107409076332632481733418240267154605919339001932537525256296703531298
224822271
q1: 105540634573358042959003671926395040098543492996728083605411385388001
181046557
e1: 5152678333963693506218275204606674519677464570076406744238955057077020
60615380722946430917712531951393382279038160929325342075983625042196382836
0140171451
n1: 1133602207508428481325077909845027734371363546274140467432860079877645
94977606746725853814780301858842395515000574491337884368940934522758863145
55101470947
d1: 1106528861132780360173000166223224996568313072049771557173432519125430
74195146952486403982324157766859859602114149454771759294840156059968881847
58332217051

Повідомлення:

54608494191133254204191010459687537937547747802253581510660032554757267041
29022260431786719819563166932241887641750808478068153235156725458588008870
99531

Шифрування:

72257147056695292685148107745900288559707422510189891249517527329427905923
85207314651606465427712229553851275947472391250922567248413486284662891617
370165

Розшифрування:

54608494191133254204191010459687537937547747802253581510660032554757267041
29022260431786719819563166932241887641750808478068153235156725458588008870
99531

Ф-ція Ейлера:

81212647658064341477393367538771555649744466981988938920288855262216439782
99980650474046062626318126270942542112028545823153452058582570722031921429
055028

Перевірка тексту: True

Перевірка ключа: True



RSA CIPHER

Cryptography > Modern Cryptography > RSA Cipher

RSA DECODER

Indicate known numbers, leave remaining cells empty.

★ VALUE OF THE CIPHER MESSAGE (INTEGER) C=

7225714705669529268514810774590028855970742251018...

★ PUBLIC KEY E (USUALLY E=65537) E=

3943901865155384478513717945613135129451652215247...

★ PUBLIC KEY VALUE (INTEGER) N=

8121264765806434147739336753877155564974446698198...

★ PRIVATE KEY VALUE (INTEGER) D=

6057692434392315235341543123332053458438720918959...

★ FACTOR 1 (PRIME NUMBER) P=

1109056885871486409645391802447405809967805671564...

★ FACTOR 2 (PRIME NUMBER) Q=

7322676473375683627772062163699423732620094408076...

★ INTERMEDIATE VALUE PHI (INTEGER) Φ=

8121264765806434147739336753877155564974446698198...

★ DISPLAY ☐ PLAINTEXT AS CHARACTER STRING
☐ COMPUTED VALUES (C,D,E,N,P,Q,...)
☒ PLAINTEXT AS INTEGER NUMBER
☐ PLAINTEXT AS HEXADECIMAL FORMAT

► CALCULATE/DECRYPT

Search for a tool

★ SEARCH A TOOL ON dCODE BY KEYWORDS:

e.g. type 'random'

★ BROWSE THE [FULL dCODE TOOLS' LIST](#)

Results

⚠️ ✓ Déryption using C,D,N

546084941911332542041910104596875379375477478
022535815106600325547572670412902226043178671
981956316693224188764175080847806815323515672
545858800887099531

RSA Cipher - [dCode](#)

Tag(s) : Modern Cryptography, Arithmetics

Share



dCode and more

dCode is free and its tools are a valuable help in games, maths, geocaching, puzzles and problems to solve every day!

Висновок:

У даній лабораторній роботі ми ознайомлювалися з тестами перевірки протоколу та методами генерації ключів для асиметричної криптосистеми типу RSA. Також ознайомлювалися з системою захисту інформації на основі криптосистеми RSA.