МІНІСТЕРСТВО ОСВІТИ І НАУКИ, МОЛОДІ ТА СПОРТУ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені Ігоря Сікорського»
ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ

Криптографія

КОМП'ЮТЕРНИЙ ПРАКТИКУМ

Робота№4

Перевірив:

Чорний О.М.

Виконали:

Студенти групи ФБ-81

ФТІ

Казначеєва Н. М.

Міснянкін В. С.

Київ – 2020

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

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.

Завдання для 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 1 p , q довжини щонайменше 256 біт. При цьому пари чисел беруться так, щоб 1 1 pq p q ; p і q – прості числа для побудови ключів абонента A, 1 p і 1 q – 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) , (,) 1 1 e n та секретні d і 1 d .
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 и 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.

<u>Хід роботи:</u>

отримуємо ключі для обох абонентів
Alice:

Keys for Alice:

Private:

272483417061490943803795784490479807015013514837593385651423172953990873935398745694547154285705608243922376556918071737170771740619522728725669623077622 3

Public:

688647269936299189342124474357034074670011043133698500385665707549842777185828549702587102882803115462664368224947100155575523495011025015946775910681793 5

Bob:

Keys for Bob:

Private:

10156729539658383716741927725327545385106094389201715120325089811215333962793267586202060420467242047034936447085129749260706085925731968152733456451887187

Public:

828043081877738493202577864240168674239372478824653602108751310744064978938941171791897351263440686348918160555199559493382951301227355358272015060628023

#Шифруємо та розшифровуємо повідомлення Аліси її ключами

Let`s encrypt and decrypt some message with Alice`s keys:

Message:105818479536079848798272986202227942438808818800758381805067164436821606470567

Encrypted message:2507336573547937404244698967629097561986968399647069764967899709841608198173165755123173942776901422341907683352153193802437004793560039833903994144323567

Decrypted message:105818479536079848798272986202227942438808818800758381805067164436821606470567

#Те ж саме повторємо з Бобом

Let`s encrypt and decrypt some message with Bob`s keys:

Message:86146148823764528447245204026188898719623924823957822820483350102483987974463

Encrypted message:47352470438721935802231782194528061860173201278368478716889239860515683496718697141525431165461824605347825127493491100290331170289147544342538446975664

Decrypted message:86146148823764528447245204026188898719623924823957822820483350102483987974463

#Шукаємо цифрову сигнатуру Аліси з її приватним ключем та перевіряємо, чи було справді надіслано повідомлення від неї

Digital signature with Alice private key:

Signature:546351062785457837555602647533521500265236917210144494563685538345637564779834533128733319489352855905979766557801049687734377213561966801098348204777 0238

It was from Alice? True

#надсилаємо та отримуємо ключ

Send Key, Receive Key prot:

We shared this key:
73419340461927584085998981593055014093161639265430015814875766217743647396159

What we have after encryption:622546590072011604064199446785053297510019791144094914397294637837896959199373593212187301387847763212188772730533906664375194080283637876236755997600 00094

Signature:319884068643752101746237141823947990876466686965324228946761521313603514795933306589043104922989178891506118358025012266266878677607964667566498461861 0342

Key:7341934046192758408599898159305501409316163926543001581487576621774364739615 9

отримуємо дані з серверу

On server we get:

Server public key: 65537

Server modulus:
460570039379734603895190075873317365696495722290002143061181172093858439665 43

Encrypted message from server:
203293507104330648581473354275679275183146981683032117467979159531064469719 93

Signature from server:
209977020196914710904795926349942273376678840527907590374666223354602511312 2

Висновки: під час виконання даної лабораторної роботи ми отримали навички роботи з RSA шифруванням, генерацією та перевірками чисел на простоту.