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

ЛАБОРАТОРНА РОБОТА №4
«Вивчення криптосистеми RSA та алгоритму електронного підпису; ознайомлення з методами генерації параметрів для асиметричних криптосистем»

Варіант 9

Виконали:
студенти групи ФБ-04
Мартиненко Денис
Курченко Максим
Подвисоцька Ольга
Перевірив:
Чорний О.

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 p1 , q1 довжини щонайменше 256 біт. При цьому пари чисел беруться так, щоб $pq \leq p1q1$; p і q – прості числа для побудови ключів абонента A, p1 і q1 – 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) , (e1, n1) та секретні d і d1.
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 $0 < k < n$.

Хід роботи

Реалізували тести Міллера-Рабіна, піднесення в степінь за модулем за допомогою схеми Горнера. Інші функції було написати просто, завзичай використовувалася схема Горнера один чи декілька разів. В якості e брали рекомендоване значення 2^16+1.

Значення системи RSA для першого:

n = 0x27da621657a82cbe53320287b3e1a92edfcf24edbb7ea91f522f9d6e585cf90c2ea9c73bd6c506c15d5f88677e2a1ae45989bcb0d11dcbdbf8174e829d653e43

e = 0x10001

d = 0x228a1b5e9fc3ea5b5d41477e13016d9e696c5ee6070ca448b7f2fa6f6af7d103f1fb31ab2f6c42e5731ed20d5c93b13b3bf644c3cc434c4eb3001c7b72544ec9

p = 0x5200e5b04d1af86c9131ab72cf0a446311304450c2aa278ae12c9aa972704d95

q = 0x7c69fb3181f10aad616653da53fea0341b368154a053044972d4d6186d14f677

Значення системи RSA для другого:

n1 = 0x16208c2218d06671558553d92ea4a50f589e14b60858f06d472f3c9c8667df23d2058687d5971f803f651c236fd731d6a88cd2fb66b334788fc27e71b5bb5445

e1 = 0x10001

d1 = 0x984f924d62e1a5499e961586763b80b712199617317e85feb271c596ecd934709eac990cd00ad1a976205340cb82c6604769137dbe6121150570f41396d0ea1

p1 = 0x2dc17ecee27b5d71098e53e5e08be613a59c251e39211974960659fe15aa3789

q1 = 0x7bcca7ed557b58ec8c88e6a36f0f370a52f8c34e61b7348699b24476be128bdd

Повідомлення від першого до другого:

message A = 0xafafafafafafaf

encrypted A = 0x1019cb73cc6201cab4b6e1605516a48d90c00110dd41af1fc271129640df370c0b2619700423b718ab5e8187610c93cc68e99715ed4d1232ddf87dac8cdf0b19

decrypted A = 0xafafafafafafaf

Повідомлення від другого до першого:

message B = 0xb3b3b3b3b3b3b3b3

encrypted B = 0xc7194ea2684022c46527874f7802b32c1882cea2862d22678d2ccb85f8f4519a23a3d7be809b3c482b0b925a9afb1e443c520d6425ea1bb6e91f57ffbb37266

decrypted B = 0xb3b3b3b3b3b3b3b3

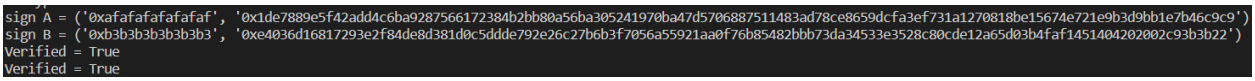
Цифровий підпис першого:

sign A = ('0xafafafafafafaf',
'0x1de7889e5f42add4c6ba9287566172384b2bb80a56ba305241970ba47d5706887511483ad78ce8659dcfa3ef731a1270818be15674e721e9b3d9bb1e7b46c9c9')

Цифровий підпис другого:

sign B = ('0xb3b3b3b3b3b3b3b3',
'0xe4036d16817293e2f84de8d381d0c5ddde792e26c27b6b3f7056a55921aa0f76b85482bbb73da34533e3528c80cde12a65d03b4faf1451404202002c93b3b22')

Через вивід в консоль бачимо, що підпис підтверджено:



Надсилання ключа:

Ключ виставили таким: 0x50505050505050

В ході обрахувань змінні набували таких значень:

S = 0xd0136ad09fc42d7a2743c12facccbbf61bd8920bb4ba0e9d00f80c342189f6d1b834d66c40b5bc06e795dd2c6cb8f20aa8f6b07ca1c54a3781e1e11a67349ef

S1 = 0x1915851878a0d97fa7158fcc2781090a8af408e301ea227c283ca6faf523098f267b6f8df5890f25455e8f7db6de6dd97d0566c1e10c64222e0fe1c9d189b3a4

k1 = 0x7d7bd821da7649d703a458097bfcc59c447316caa71685870ea9048ed6845de618ee3315675a9d781789893e906a33d6cbd54018c5f30598a960951d37557d7

k = 0x50505050505050

S = 0xd0136ad09fc42d7a2743c12facccbbf61bd8920bb4ba0e9d00f80c342189f6d1b834d66c40b5bc06e795dd2c6cb8f20aa8f6b07ca1c54a3781e1e11a67349ef

Як бачимо, значення S, k рівні, тому протокол працює праильно.

Висновок:

В лабораторній познайомилися з принципом відбору чисел, кращим способом піднесення чисел за модулем, системою RSA та як нею користуватися для цілей відправки повідомлення, цифрового підпису, спільного секрету.