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
Кравченко Владислав
та
Жмур Назар

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 ймовірнісні тести Міллера-Рабіна з попередньою перевіркою подільності на деякі прості числа. Реалізували піднесення до степеню за модулем за схемою Горнера. В якості відкритого ключа обрали рекомендоване e.

Значення криптоситеми RSA першого:

n1 =
0x2c1b756b83757b33c59da3605bba322c06d41f88d397805f405ba3474e7d40f974c00b0f251cbdaf96aa6fff38639a7cf7236b2dc54dc99353a269f4bad72651

e1 = 0x10001

d1 = 0xb3979aa40ac170c57d00411b38630da6f6a7013fbf4e420595a78b4dece954820f3cba3470117d4716af7cca7ea593c93cd4971e8b8f3a8569482d09c190981

p1 = 0xddfdfe0f686803815e4ad33411791126de3ee0766c35562e48438946958421c9

q1 = 0x32dd3dbb9341a77611554f40db59a64d5500f33a761fa8c2bc07f27016bf6449

Значення криптоситеми RSA другого:

n2 = 0xd24422cfed449464c2ec3978aa3cf9cd864f283a4781eb6d45431a8712f0d147adba6ebcdb84d0dc30478a635b9fdfd73bf23da47d4967449478af550ca685

e2 = 0x10001

d2 = 0x98fa4b9d9d9174a96e8bb1309222f4d7cbecd166bd59cfb05b60fbeee41fe262511c3fd8c46447a357fcd6bd892a761ace4cb86b000834dae9cb6e849e6481

p2 = 0xde0815cb9493acd798135b82390bebbacd338666f8ae2bfff5f0703a85ffe5

q2 = 0xf26f4160d2f91e2a439e38b5edfbceece4c7d48977308568361ff351af7d6221

Повідомлення від першого до другого:

msg1 = 0x7734773477347734

emsg1 = 0x1444d1275e411b061bb86870f564a146f1e200a91f3b921bd52bda2f9cdbcee4dad83de229feda2d40ad5602fc549350deba0fd7dcc83de0f482d5c5ae15e5db

dmsg1 = 0x7734773477347734

Повідомлення від другого до першого:

msg2 = 0x5100510051005100

emsg2 = 0xc00d8ef3bff8bd4c8c4028611a5617febe8aeada5ae35a6bb21da8c529d27dce27730c5892f07556c80bae197afc6be54a3611af5e1a999f7db09f4e1f949d

dmsg2 = 0xb3b3b3b3b3b3b3

Цифровий підпис першого:

sig1 = ('0x7734773477347734', '0x1463f6768b3652dd0f13f2be652d98b102b5351d9d1f4755e60ba105bf8f82b2a4e13bd7e145f1397cc0fdbf7b3983aaee6d51974149b1e8e7762f7a6628db5')

Цифровий підпис другого:

sig2 = ('0x5100510051005100', '0x607c72d4c0b63831446ef9e9f54f61285514a953f551f9edb7b9de3b764dfa20184f652725c7270df21e799d3293a4c2bde7dce52580a5d910e2d8358b6333')

Надсилання ключа k = 0xc00deedc00f

S =
0xd1b8f3cf02acd9fdb58914f261c4555baf55e263a3d0154a2fc760575bb602a72a3c2c03e1675ab092dbfa7d2315c0a3a24c8f3b0ecf2177395b2456a060c5

S1 =
0x17dd48dc79521c8627eadfccfb0a4eb3edcd6894eb6ab976f24c94168b9d1d684e19fe2071ee076b89f29f99d2972becf76679f029971a330f4115725aeae78c

k1 =
0x20bde9e189860a8a24ebc9cfa66ce7eba440175d327339a3b7d098988e5253bbdff10749693f781621353027057053f74e81e02ccc6c104cb9d69532af5c2357

k = 0xc00deedc00f

S =
0xd1b8f3cf02acd9fdb58914f261c4555baf55e263a3d0154a2fc760575bb602a72a3c2c03e1675ab092dbfa7d2315c0a3a24c8f3b0ecf2177395b2456a060c5

Як бачимо, все зійшлося.

Висновки:

В лабораторній роботі познайомилися з підбором великих простих чисел, їх перевіркою, функціонування криптосистеми RSA.