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

Виконали:
студентки ФБ-06
Товкач К.В.
Вєрнікова Л.Г.

Київ – 2022

Варіант 10

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 $pq \le p_1 q_1$; p і q – прості числа для побудови ключів абонента А, p_1 і 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), (e_1, n_1) та секретні d і d1.
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 М і знайти криптограму для абонентів А и 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 $0 < k < n$.

Хід роботи:

На початку ми написали потрібні нам функції а саме функції для знаходження простоти числа, перевірку чисел на елементарну простоту, підведення до степеня по модулю.
Далі ми виконували найскладнішу для нас частину, а саме тест Міллера-Рабіна. Якщо слідувати коду, спочатку ми написали допоміжну функцію s_d для знаходження параметрів для тесту. Далі для реалізації самого теста ми написали окрему функцію, яка є циклічною та містить багато перевірок, згідно методички.

Наступні наші функції основані на вбудованих бібліотеках, і слугують для отримання випадкових чисел, необхідних для реалізації алгоритму RSA.

Потім, переходимо до суті завдання і за допомогою функцій ми будуємо пари чисел і генеруємо ключі.

Пари які ми утворили:

p = 642838629492478957369158784399613817184966745387618307544780328524324712933 29

q = 588709217582325549106123250815065053236764169645021003509814830497170361017 51

p1 = 817119964042975855225173836467748650421897275328991782222387789693009903942 83

q1 = 713110067294133178966824938400164838303453689167657410661525676275905375758 21

Використовуючи знання модульної арифметики, теорії чисел та фрагменти коду з попередньої лабораторної для нас не виникло проблеми реалізувати функції потрібні для нашого коду.

Результат:

```
Ключі для абонента A у HEX форматі:
e : 0x26e8797a8eddf09d16f79a191a876f08ca7c3739959cd076294d1ab07b0e83e9338778c484f9dd01de32de214b4a5744b91b1b61bd82f497d27b931eb9bd40f3
n : 0x484201f057000f1de13a9efaabb20816f0e2c06abff80d155b0b3ff2c74852b405f8ffbfb0c5269c56eb3d357c60f47267d9fbbacdd138f209f3e6065c0e8e67
d : 0x2a861daca5e6e86be1bf985259070d6f9bf10b2efbd95d5e83f163d5314b37d6882242f8c2a8b0f3470a6a642cc0aefae714b10036e31b345a38046452d8b83b
Ключі для абонента B у HEX форматі:
e1 : 0x1f0d087eda1af0d44cde342d6682a5e9ac57c21a5e031a7da55cc0d5c501261844bb4138ca4b961e3c59c137df2d5daea322ea50af8eb964c65ccaa7f6e3f11
n1 : 0x6f419c3bb11241576c17c3b93c00e3c08850e23366f01310bbfc8020935b719342148030eaedd78d0a77aadd8e17c44b36d2f86652645b13dfd8455a3bc6502
d1 : 0x6b6488916716bfc8980e3bdcb8178d72f13e2679c7e019878483b1b54b8c62f82adcd66212c0c88c33e65671b13652288f0979579113b016c945664d94ed3ec
0x25a12786029a0d561c2ae15f27aad20837e57b2fbe44676288ee45ae4a8e093c5de565dc087c134d104962656e96e6f8cf9367e09ba96233e41f5e1cd0c0130b
0x43d599268d6dfb87c636f7963a66f528a893710648330892305800663d244026e1d62aba98afdd340a118c823351d8228724c209edf3af6753bb891c7a80412e
0x25a12786029a0d561c2ae15f27aad20837e57b2fbe44676288ee45ae4a8e093c5de565dc087c134d104962656e96e6f8cf9367e09ba96233e41f5e1cd0c0130b
Перевірка підпису вдала
Аутентифікація вдала!
```

Найбільш проблематично було описати алгоритм тестування Міллера-Рабіна, так як він досить заплутаний у циклах, але завдяки теоретичним матеріалам нам це вдалось. Та, так як RSA досить новітній алгоритм – код треба було оптимізовувати задля більш ефективного використання ним ресурсів комп'ютера.

Висновки:

В результаті виконання лабораторної роботи, ми ознайомились з тестами перевірки чисел на простоту, методами генерації ключів для асиметричної криптосистеми. Реалізували код який побудований на основі криптосхеми RSA і створює засекречений зв’язок двух сторін з використання електронного підпису та секретних ключів.