МІНІСТЕРСТВО ОСВІТИ І НАУКИ, МОЛОДІ ТА СПОРТУ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

ФІЗИКО- ТЕХНІЧНИЙ ІНСТИТУТ

Кафедра інформаційної безпеки

КОМП’ЮТЕРНИЙ ПРАКТИКУМ №4

з дисципліни

Криптографія

З теми: «Вивчення криптосистеми RSA та алгоритму електронного підпису; ознайомлення з методами генерації параметрів для асиметричних криптосистем»

Виконав студент
групи ФБ-91
Братунець Дмитро

Мета та основні завдання 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 і рекомендації щодо 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 pq ≤ p1q1 ; p і q – прості числа для побудови ключів абонента A, 1 p 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) , (,) 1 n1 e 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. 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().

Значення p,q,p1,q1

```
q = 819944755844059359488689559116302771823895304354025390018169167203922627708 79
p = 606950601254063536452055869781130495194236576426920927004807836118282810084 89
q1 = 863214809492125748634412233308077493185022280605287667664501792063382120094 11
p1 = 814853292241268696312106654573375540191046615394557103713434497879893402450 59
```

Кандидати, що не пройшли перевірку

```
832056463405735883517827347735463826826238840115023342150883231625198359698 44
779955684022473715875466812538582352649208813514992704510218884648779830216 20
107040087594361302076408804455307662764024165418927812341303184551920807605409
782790341477737280937158474709950345193614441668834920976214150087664445859 13
639278258500429017858131657018117175652011324732374069334544117041941450110 34
115443413465751963941219313348089100444991155009394978338570913975124513578903
101413398792510583534762628335542382870175266137993821599215896803885073952785
102458591876811878086073371831904007300417517513196473500515133214678179220025
980091183042218108385697941763294620104748073164999897497023404956296002538 43
104041065319178400334649349075318522911193206045180626800195025303666333677482
929280260286178405923296832332645439398875778189027666894108479045013612890 70
104165251172790549567829559567230661952062409417221742982757295226827475745651
897785148372937529458595999962394610057337929558048583243004245762613694428 03
112334507711927524321043464258435528099622029358791566574472688498099162798629
754228817318463530891313791701291667283305921923596978326296143651728996708 75
810448887284371599461093560601766471335864391394601185659145190107528106749 63
722477244609114912527145153700326046006965591138386212868374320667435730294 72
110340483148123874130766370153156891309335888568466323259766810057337128169518
814089787127733314969405405887191256160911026661334563278988421893571944924 76
846332794899149846107655860964655747847773687892713204152400173443755752271 95
775630225670438823399539031106840175546577041224533966534777714038250864124 61
936305391378285677565400881784348492578118241755393801598145767524171370622 99
779858838437435953806710272408495868490809989559968506028045629778813880371 94
111370875257756427421692687096149238796966774408431699996223756347568768289230
```

…

d =
958823096631433050497324459207640072391570979734266255523453993993723654231759941008368950448179037447997499231471844206731735982851343697230597811522579

n =
497665962554668154886394405849219553057478485433621436542088936773116792144143071651531549541006777866938945291617965666233766909925110110360534766099183 1

e =
398486176378821262750877995584578931674681652098576197747244272109853596483780555419036765916156139265023784724700525288686941076652134312452823026886266 7

d1 =
545055688627887281868541163446661030632002622533526181693417010278093849083671068676156903526307995404368453470533160889201652207386388291635851209608906 9

n1 =
703393429426078226076184000805988480216601059469805443949017699570516933609724173342708201997010812675634764882091013908218190558158716274088357555425024 9

e1 =
537007001075206016457856478164813525238802985825363653225494441253951190984310365734513898609894709629926094149261988724784471366685138966112776921770938 9

Відкритий текст:

m =
48372797596149753898932491379147483525246561175484473497755577621036487936981287020350362237236259505514627689783298605583778117059832230009700375560116 51

Шифрований текст A:

c_A=
81051239050998055129755753393956046163130786245145441260287370274470721291174082332524703327418140308938691604069445381681143944208961561949462214688667 0

Шифрований текст B:

c_B=
25930476839888539624503111250387218748976740930978689883812289305488869564568090889694419637020309003146519812891890672315293330814397798241200358814664 66

Підпис A:

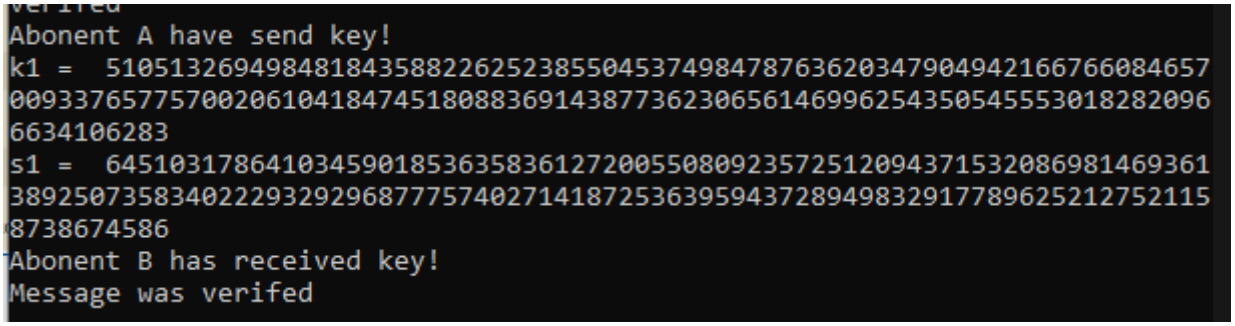
sign_A =
37713331145484627181675319256972933021664332499598618790874707831064338780197315302826893906120950655220321212184077502403287537207217163827423061761541 75

Підпис B:

sign_B =
31987508439680777851489808357843684291214334190544949959433001809008170482707617178474000355891242036173020270167265960950465356003020626060288916721420 77

Для перевірки роботи криптосистеми використав сайт https://www.dcode.fr/rsa-cipher

Протокол конфіденційного розсилання ключів по відкритих каналах зв'язку з підтвердженням справжності відправника



Висновки:

Працюючи над роботою я навчився перевіряти числа на простоту за допомогою теста Міллера-Рабіна та побудував криптосистему RSA. Ознайомився з методами генерації ключей та з системою захисту інформації на основі криптосхеми RSA.