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

Комп’ютерний практикум №4

З дисципліни «Криптографія»

Виконали:
Студенти групи ФБ-91
Мельник А.М., Тислицький Д. В.
Варіант 12

Київ – 2021

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

Мета: 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.

Постановка задачі:

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 $pq \le p_1q_1$; p і q – прості числа для побудови ключів абонента A, p_1, q_1 – 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) , (e_1, n_1) та секретні d і d_1 .
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().
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.

Хід роботи:

Обрані числа:

p	204901066145616790514004106998368362225858044793392992932266945703983824399231
q	475082654572977601398337425500715893335301346496660089680978710628964109092793
p_1	95007513074708714572428002475778390358900370578910621481465972417194730194932 3
q_1	96104190294625260426139658634703338179321942828321480696575504706035963402849 3

Список кандидатів, що не пройшли тест перевірки простоти (в таблиці наведено лише 20 кандидатів, всі інші можна переглянути у файлі test_numbers.txt):

№	Кандидат
1	1884531040356235976608388577787060723976190444613871209819918029108526852393 21
2	5324592715257613056921271061047033914750188935375634296136361024861368612584 95
3	9365134333144708973977800960948196884250303515443751176572655642114023886641 55
4	7499124125867889829983758237732561335891295308073841263466094558558013140088 25
5	8583358316806581513951439838300177689535578238553761561625467685097889460006 79
6	3068610660667836125606375470704624398333529599250501849979200763218722890040 67
7	9112814573396791885178778028864514287275798901031240052398183601087300103407 55
8	4497862852197957079865100624352711667457882106282585082814163013767089824033 99
9	5472376360072296796931662902568688129692886846393125429053924685683644216051 75
10	9145750070129536873985757416396953064614047218106766250841577164724418777851 89
11	8329392223621309552936575048705213391893414299387181219898395351732395682343 23
12	3477841562260953003530630783240516828035928529727744625478142744805342695699 35
13	7535698752481122356360202514222513152353421759966851219629914799257354731390 27
14	3325786142294690568574292671075652757823727391780868450619937634155758055481 87
15	1416871313484408222184358334274473279842916808056527422547433516826789630250 43
16	7884414294508507895162984338725445906491194402959431447144348076844675741852 25
17	1548844183763806589488482348313046877097267751805335178758101890909390871982 93
18	6956127883993315525107544711654625831027719250890511428644326608816297548023 77
19	6044902930191973891709190997012308847201777669413674548888021700233351658565 27
20	4080670532656934197288995871143797980284612857587824301747195667189823650752 85

Параметри криптосистеми RSA:

- для абоненту А:

n	97344942429292896708432683784515333078537407546561739218560401448468381982691693789262862119487115077645389767083900437472794398352619283945572176568421 83
e	69542412498258837415532782246404634337180862035691600727250919886554844328753341357102099926688715379774458692468930429714372614993075463540739533454592 77
d	69435979944044534922039975851219677728897078242276375359869841861230808170850162126626619021350814268696884814401342057611639315999589271034420725055710 13

ВТ:

55853877372247607935678112557338729809978440155745030452287075773330625369217768964517701384207319774256827184089528075173602320573668678454296501100602432

ШТ:

42583673414514619088305703782047023663816806166239778243132719954868467461701354544604280496834185322730576339706414855704608732187406532295596946184060153

Розшифрований текст:

55853877372247607935678112557338729809978440155745030452287075773330625369217768964517701384207319774256827184089528075173602320573668678454296501100602432

Цифровий підпис:

84036223822904153795257918739008260426128319190150052078288066185025838264813799198130190591554508904309121717991302488978201659385641021572504932429233282

- Для абоненту В:

n_1	91306201159509037820003131558085516638000629163631255319553577996357090444367239267707462248012091096969750341792539201181801162034172169609173022240602 39
e_1	27372655469112712856601138729470022317109567242046562843444521879071783736332153443855152927909486454292739989373846833575011539855441169668841259402080 43
d_1	70242045485816493828049955626661552002186655554446514597454065384037576734881540149825367523173069916227967706690006187679720840252314452295857887209080 59

ВТ:

86952378368108420624758415100048835919256315456688789313295449733923695113305183018951184676925662524975080387028377477779929819139317969370158430611991859 6

ШТ:
170915869242776142229356521877385436952764120796540906228385838179210940
977883218569173927680576248111819241496002549606354440193262726568329534
967004391015

Розшифрований текст:
869523783681084206247584151000488359192563154566887893132954497339236951
133051830189511846769256625249750803870283774777799298191393179693701584
306119918596

Цифровий підпис:
275239024207651009820480525957119491480063025307804940625664857723844569
404917123579239826297283667820594196545641174429393227255360729681177278
249971535087

Кроки протоколу конфіденційного розсилання ключів з підтвердженням справжності: (чисельні значення характеристик на кожному кроці):

1. Процедура send_key в якості параметрів приймає, k – випадкове число, що генерується абонентом А в проміжку від 0 до n не включно, e1, n1 – значення відкритого ключу для абонента В, d – закритий ключ користувача А та значення n для користувача А.
   k =
   207698827643653117667442165931866971874783127169278269720558353370211 4706132374377066973255507864346755834221515754582114952671772503843953 842368995618021

2. Абонент А формує повідомлення:
   Змінна k зашифровується за допомогою відкритого ключа отримувача та зберігається в змінній k1.
   Змінна k підписується за допомогою секретного ключа та модуля відправника й потім підписується s за допомогою відкритого ключа отримувача та зберігається в змінній s1.

k_1	520010633923541047998451594497002310536843553338861808995557765326987435151350449799401040672535698625221397772149501796574922507973635009605581628628402 3
S_1	549209773630697649795916718568626225444058557184031712318909437975801659732239823618694601662112969125877823673233648463390505287390159413061028512177352 8

3. Абонент В за допомогою свого секретного ключа d_1 знаходить:

Для цього використовується процедура receive_key, що в якості параметрів приймає отримане повідомлення k_1, S_1 , а також секретний ключ d_1 та модуль n_1 одержувача.

На виході отримуємо значення:

k	207698827643653117667442165931866971874783127169278269720558353370211447061324377066973255507864346755834221515754582114952671772503843953842368995618021
S	310459179215842007827264917092025732498117653498243973510715454088690438057206334606096660883141170634550159676977186662795140179896411701255081652463700

4. Абонент B за допомогою відкритого ключа (e, n) абонента A перевіряє підпис A:

Це виконує процедура authentication_k, що на виході повертає рядок «Verification state: True» або «Verification state: False».

Результат роботи наведено фрагменту наведений нижче, всі результати можна переглянути в файлі results.txt:

A generated a secret value k:
207698827643653117667442165931866971874783127169278269720558353370211447061323743770669732555078643467558342215157545821149526717725038439538423689956180 21
A formed a message (k1, S1):
(52001063392354104799845159449700231053684355333886180899555776532698743515135780449799401040672535698625221397772149501796574922507973635009605581628628402 3,
54920977363069764979591671856862622544405855718403171231890943797580165973223279823618694601662112969125877823673233648463390505287390159413061028512177352 8)
B received (k, S):
(207698827643653117667442165931866971874783127169278269720558353370211447061323743770669732555078643467558342215157545821149526717725038439538423689956180 21,
310459179215842007827264917092025732498117653498243973510715454088690438057209863346060966608831411706345501596769771866627951401798964117012550816524637 0
Verification state: True

Висновки:

У ході виконання даної лабораторної роботи, ми зрозуміли, як працює криптосистема типу RSA, навчились організовувати засекречений зв’язок і генерувати цифровий підпис з використанням цієї системи. Ми також ознайомились з рядом сучасних методів, що використовуються для тестування великих чисел на простоту і навчились використовувати її на практиці.