Міністерство освіти і науки України
НТУУ«Київський політехнічний інститут»
Фізико-технічний інститут

«КРИПТОГРАФІЯ»

КОМП'ЮТЕРНИЙ ПРАКТИКУМ №4

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

Виконали:
Студенти III курса ФТІ
групи ФБ-81
Легкий Дмитро
Чалий Олексій

Перевірив:
Чорний О.М

Київ – 2020

Хід роботи

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 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 A, p_1 і 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().

Кожну операцію рекомендується перевіряти шляхом взаємодії із тестовим середовищем, розташованим за адресою

http://asymcryptwebservice.appspot.com/?section=rsa.

Наприклад, 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

Генерація пар RSA для двох абонентів

Abonent A:
First prime number p =
805913212210634693031595532599065514752130168230371278286655397967547
79707791
Second prime number q =
590566320565257827302736409314188944580945615997853156837166988582872
33620011
Public key(exponent) =
298638603943632677425221214060633964728125586217641041360449451032831
823363685319954841756828968716105857907446954136773085719985509260235
06243703633486 69
Private key =
261629901610770104857789258822543375003083510759064627740690687184131
410091695480509411586768143176045533951209329285821124200213435863132
44251103153189 29
modulus =
475945200430162346856751522372833409733428183073577234505444452597985
838251208095754550820653629266364413780643222543956951888348005964396
8213653710205701

Abonent B:
First prime number p =
108471423585983071553281999992091670509506424297598296128632881703862
558530003
Second prime number q =
623403431192890489455200171778741028277998766433453192237945548484366
74459851
Public key(exponent) =
217000577014311002030185729323397246753193095096207998509596924226619
696089550360754866282010952800061798475910437550393490184561312284248
188232734188021
Private key =
499237016190305904726668773177765527852227722659185329981798246701246
670351340817085713344796991219851010625928751658054411482309031536908
4331290933192781
modulus =
676214576498792763016371169726705501118064089708764983177302539386942
568475726289372631686322052757507824140956739259503886486238647840039
0975932602409553

Зашифровка і розшифровка повідомлення для абонента A
message encryption and decryption with A public key:
message =
3102698143992293786820312116075506967501356930202381120863754669643408622788 7
encrypted message =
2101778699432811812355090074564213867216801853126905449333940805289283417987620744152345403469486242382342520093295606678061710409624070177525061753121932
decrypted message =
3102698143992293786820312116075506967501356930202381120863754669643408622788 7

Абонент A підписує відкрите повідомлення, абонент B перевіряє підпис
digital signature with A private key:
message =
3102698143992293786820312116075506967501356930202381120863754669643408622788 7
signature =
9161908321564601441364404600752871804744182723635441680441515603449769956329851438300246466290143757015570874904946251127571302574525395007831235925188 82
From A?: True

Протокол надсилання секретного повідомлення від A до B
SendKey, ReceiveKey protocol:
shared key =
5712630057410002566308446956643364662916313595765720845918841783864079237735 9
encrypted message =
6075821374553631004039174742698117012173079791605722845439124665486166762235183515916687594791048935385007017335694771658010801454709371736442523703158488
signature =
5037900369911899625792325517433171053046747526060869905585379642160800228921603217481240284985307924824850103022548526506824935230008572494188428072602439
key =
5712630057410002566308446956643364662916313595765720845918841783864079237735 9

Протокол надсилання секретного повідомлення від A до тестового сайту
We send key, server receive:
Server public key(exponent) = 65537

Server modulus =
19659984280503115671862636730975359271135747328483649882392121862145
52228501443808049752092403110537714471104179675100285081362003977257
328742762593882267451
encrypted message =
80752103355026456242762455984819555125471664285948982939109703804641
18438736608840883336326807686975502966958389253883511119098697360284 9
5252644256089378882
signature =
17571957721310642324694506124824686036512686143564723721242548113997
62670457487624740998134060296546974314166790710234082877571728568168
508362061955374662411

Переходимо за посиланням і дивимось на результат
http://asymcryptwebservice.appspot.com/rsa/receiveKey?key=3c3a48976c1f8dd1f01
7463e2e5407463e56d4e4534d8043023801d6687f9d4d5a46b41911c23e23877559272
d986f6ed60baf63d4b2dc9c655a8b5a9b6e9c1042&signature=830ec204f1bb2fdd7cb8
40ab4b1453b7ed80b14252635f7ac6b95d2bb826e37430b513221ff9f09a54e6c3dea3f
eb8c43105cdde742bca7f352e79e718183a430b&modulus=5adfb780060bc75faafdd83
a95aef466e684589f83b3439490501d5db2dfff87423df433446fe4d1136ec4581f1a073
657da4b9c3079627b233956d187207305&publicExponent=390526a0d6e34440bad0d
f2dc07e56f6133ce9d6960f436efe7a312d10aa47d2ee735e0bcc3deb868fbd7b2567795
f97f10e452c0bc6f7973aeab5b6c6c236bd

{"key":"75B9F560069EB461F5550D56331F3D08A4BA0E3DCC471CF3BA721572F6D3A877","verified":true}

Висновок

В результаті виконанні лабораторної роботи, була побудована криптосистема RSA з генератором великих сильно псевдопростих чисел, тести Міллера-Рабіна та тест пробних ділень для перевірки згенерованих чисел. Реалізовано отримання сайтом зашифрованого повідомлення.