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

Криптографія

Комп’ютерний практикум №4

Виконали студенти групи
ФБ-81 Скляр Б.Ю., Висіцький
С. І.
Перевірив: Чорний О.М.

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 1 1 pq p q ; p і q – прості числа для побудови ключів абонента A, 1 p і 1 q – 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 1 e n та секретні d і 1 d . 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

1. ***Була написана функція генерації випадкових простих чисел заданої довжини. Перевірку на простоту проходили за допомогою теста Міллера-Рабіна:***

Generating key pair for Alice:
960237818602862526494690584556912838739103938858395368121790921408076
35532331 Is not a prime number.
960237818602862526494690584556912838739103938858395368121790921408076
35532333 Is not a prime number.
960237818602862526494690584556912838739103938858395368121790921408076
35532335 Is not a prime number.

2. Сгенеровані p,q для локального кейсу:

Alice

Public exponent (e) is
0xf44318a0b2922f71e2d6375f729b07cba824e87fac8c530f1d24babb6382e868d25896
b648ee2b9e71a5b9cf413834da84b6cc10ea0c739de52597b95d659ab7
Private key (d) is
0x9c82ab2aa7c4b5c4ca35324ff9cee3ccb01444dc759fb67920a951d354fb61f74f6e617
f6b86f5cd46ae97998ce8013b7110e768295f362aa875533f2577717f:
Modulus is (n) is
0x3a201fd1c317e99b4e7ac1853569bf04a8c43fa6eb68388fad5f3e946fcee1df293ba05
71659fd17fe81e09bb6bdcd6f1e5bf7b59486dfb6c744aad417ec90ee1

Bob

Public exponent (e) is
0x75acb3fbdff7844a15a808f85a32ea6edb1fffe8a1072f639fe60e39ac83cb2b9b4dd34
dd0a31f7f163ebc8831846c5b0d62d0c454d3cc97c02001abf60f84deb
Private key (d) is
0x315fd5f34d6005a0ac8852f657ab17f7711b136623571a4e6e66000ac75a8d289ab91
32af9287e1d189db84fa6f80e87af08c2105c84ed87333969820ff878253:
Modulus is (n) is
0xa3de0ee50afae0c60821c8c417d44051ccb189c610141866391ff4a40a566627e97a65
59d11a67aa84e44292d93e50efaff735b972ea2545b0647d8f464f6d34d

3. Протокол Send та Receive:

Shared key is: 124
Signature is:
398247903152598959509934809234765304863526648383207653736608430999386
570940433302083110607565919639316807305496258411641273282599093658105
23092185250419709
Encrypted signature:
864453711124775642908874867352755957529263236950089189273690239666811
187677224041035065153348413564107959890199398336983432480016731824651
49738040105279313

Encrypted message:
13525674860654182442807005936657979552696882997846974702385117824856781759069692581059343788316275485959109960024445182246836042396460279999708249809949987 5
Starting to receive a key
The key is: 124
Signature is:
39824790315259895950993480923476530486352664838320765373660843099938657094043330208311060756591963931680730549625841164127328259909365810523092185250419709
The key was successfully received

4. *Протокол Send в реалізації з веб-сайтом*

Secret message was generated: 0x7b
API public key:{'modulus': 'B8185C023688F8C4C2C77188A046F8F508C7B1B4281783C8AB871549A53A7130976E14256B3C9B764FEB26C14D026E5CD050E0D92929E4F64B68E59D7B3546EB', 'publicExponent': '10001'}

Site has generated an open key (e=65537, n=964184556588845838234590471880345704640517574900015941200953254975258209792756075752702202188559583348500448998162693455007654665442596844392279815050212 3) for Bob)
Signature is:
55701379861824525189745982063587621974209647666512619249830254387425352270053743418885449176602044882256927916783804489736332311435421096490407223387682308
30010620008604876344812311659905795786811385397589761038090247870370068938460711119774130754501604928016032217177499444902260853578543910389190908085275823
0x394ce2afec4df97596d15b98c43ecdd1f32808ca61c99aed5bf0b74aa2ce32fbf8ba05949a4e956845fdebd065a938454199d16d2b0c0c1f52fefcf7a0991caf
0394ce2afec4df97596d15b98c43ecdd1f32808ca61c99aed5bf0b74aa2ce32fbf8ba05949a4e956845fdebd065a938454199d16d2b0c0c1f52fefcf7a0991caf
Answer: {'key': '7B', 'verified': True}

Висновки

В цьому лабараторному практикумі ми ознайомились з тестом числа на простоту (тест Міллера- Рабінсона) і способами підбору простих чисел великої довжини. Реалізували систему захисту інформації на основі криптосхеми RSA, а також протокол конфіденційного розсилання ключів по відкритих каналах зв'язку з підтвердженням справжності відправника.