Міністерство освіти і науки України

Національний Технічний Університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

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 групи ФБ-84

Мороз Дмитро

Яловчук Михайло

Перевірив:

Чорний О.М.

Київ — 2020

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 $pq \le p_1q_1$; 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.

p=1085c77a281cf238844f9299aaef4a0fe13591920aa784f74dffbef3341e1060f

q=118ea4d3cea374733a86837ad8bfa8d5fafe76d25039dbd3d6611f6a11fbdc331

p1=128c9e38dc5ab36d31bc8e46e8fbbfe49fd49bc170a5913ce7f0ce2a697ba9cc1

q1=1c22a770b7bd68ad2dd331e0a7dbdfcfbcd005c5502a508b47a603af1be71d8cb

n=122171721d00abc147c4f50de270c92d4b4c3d415b2c14c02ead763626ddb9921da1ababdbb
b3e42a95093b3a6e5095ee7672ca6a1234586b9f94f04a4dc295df

e=b1b1cd04d24b3231ccdc6cb90b6706c3eb05ecd4033abef061719c8ad773e11a43ed4ba947f4
9da8877452db6c6f7beb95600a5cd1747f21c9f902748e118d49

d=93cd728d99f07ac9f8731cb697c55f3defc2b5dd99d5f86c9a5c15442ff732e7fec6b358c44e83b
574e9287bc6d4af057cbae2c935b57ce80fb76af42e478df9

n1=209e41d21b3becc7acb86514ba6a84bd5560514e5019033ce3e31d087abcadafdd8618cbf20
4da85f209534588234d33e0f58b682558ec6712f2b0f87346e250b

e1=5d655a62feda48432013b7e2b98f3ccef9d72491012562060b7dd9b9a40bdde26855c2359fe
27f8544d79305626580b5a3ec1313f999866d55035e04661495a7

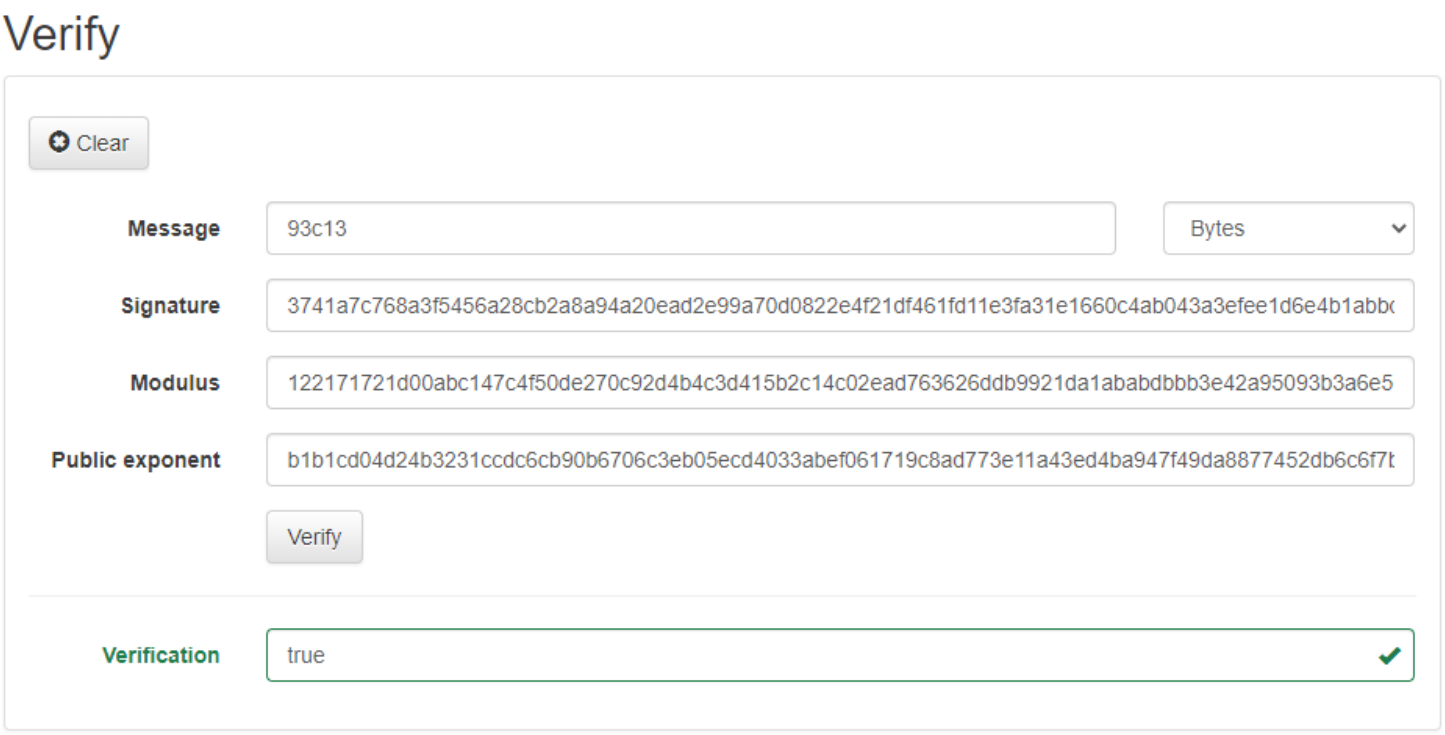
d1=ffc16c1d38e08d64057ac11302c8a5b5d952bc81fd56ef3563d87a9be9466792af45a7c1a13a
6f4b32e19160e7d522186384508de58070beb61651d354780d17

Чисельні значення прикладів ВТ, ШТ, цифрового підпису:

ВТ: 93c13

ШТ:605203

Цифровий підпис:3741a7c768a3f5456a28cb2a8a94a20ead2e99a70d0822e4f21df461fd11e3fa31e1660c4ab043a3efee1d6e4b1abbc3028247c9e4572d62673f81f112b458aa



Висновок

В ході лабораторної роботи ми ознайомились з тестом перевірки числа на простоту (функція Міллера- Рабіна), і методами генерації ключів для асиметричної криптосистеми типу RSA. Ознайомились з системою захисту інформації на основі криптосхеми RSA, організація з використанням цієї системи засекреченого зв'язку й електронного підпису, вивчення протоколу розсилання ключів.