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:

Студент групи ФБ-84

Чипчев Дмитро

Студент групи ФБ-84

Киричук Тарас

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 1 1 p , q довжини щонайменше 256 біт. При цьому пари чисел беруться так, щоб $pq \leq p1q1$; 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(). 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 та опис труднощів

Спочатку було створено функцію для перевірки числа на простоту (Міллера-Рабіна).Після цього було сформовано генератором випадкових чисел дві пари простих чисел p , q і p_1 ,q_1 (256 біт) , такі що : pq <= p_1q_1 , після цього було обраховано функцію Ойлера для чисел p , q , далі генератором випадковим чисел було обрано число e , та за допомогою функції reverse() було обраховано d.Після всіх цих перетворень було отримано відкритий (n,e) та секретний ключ (d,p,q). Було реалізовано функції : GenerateKeyPair(), Encrypt(), Decrypt(), Sign(), Verify(), SendKey(), ReceiveKey().

Вибрані числа для A і B:

p = 93e14c6f15b282d9c4f52c85da3d9501d8655f9bfff6dcc9be4c7126d30a74f1
q = 94de7c27b57e49c25fd4d89aa89d2fd48fead9b57bd2b5c7063b96654ea09b27
n=p*q=55fec548920fb66bd43101bc6da89778da43ed1246e57dd6bf021182036d129ce60d
d97717b4f09f303be06d44d8b3a0ea04b238f23795abcfe92df10005bbb7
e=35692cbc18efc8f32d5f21de76261f391b68cad933b02e9e719391b16c405903960bada2f
0b97b5262b8caed0a3f5f5188b31032e90b3794cfd9af9b76549c0f
d=2e7be77ecfaccc114f39ae7f04fe22970155a435bd3ecd68b60722c5f9adff6af2e65952a69
2fabfd90ad329c2232f9872952db2ee1e9431ff7a2b1af13e660f
p1=f761ccd833f1688c5309e6e124b0da2ce7a2c790a0a00d19c35d180f296c5233
q1=c4d3cd22c9acf3ccd3ef8b4454521531ded31c8afcb61ebb3d0495b7040b76e3
n1=p1*q1=be338cac4ce0e576f83d44a8294a5add1228f0c770a69f7ce0bec3b908899e860e
6dfe12d4d3446bf799a2172619e0ce61d5f00596cad7c082023ceefd216539
e1=a155fd844d0afd314caf9bc9257664c43b07d392970077e53994e2d8f617b9660ae26186
2440c64b93a6da86c96ceda673e2b6d075ecfd9d7fdd527c4d57ee15
d1=b42f1c4303dcc611d4115543eb9694bbd6e158db75f88f4b38c3dc1c9aca8e8e10862cc1
a1c80a92628b37e9721ff897a4bfab35c4804f6d26b43e0766319d21

Чисельні значення прикладів ВТ, ШТ, цифрового підпису:

ВТ:

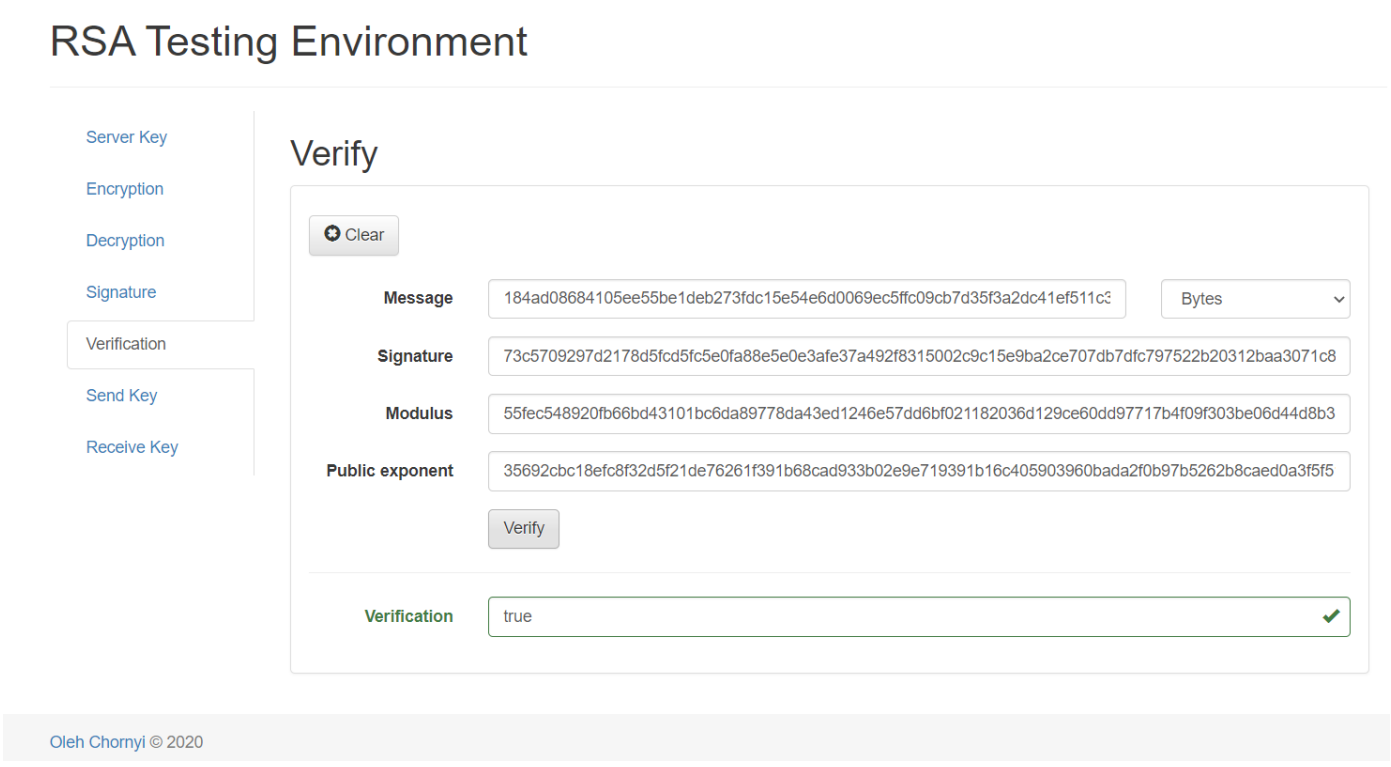
184ad08684105ee55be1deb273fdc15e54e6d0069ec5ffc09cb7d35f3a2dc41ef511c30f9dfa70f956f0ca6096c6f9f5954bbf75298e62a5bb90fd24ec9493b6

ШТ:

63054b56bb431d2b1b1e68898b1777ac8eb029a3b9a4db8b3a04e21c273c2fed64ff319cfa2573b2870810b40a072380b1d643534b402d0fb2e0bfb54b49afa6

Цифровий підпис:

73c5709297d2178d5fcd5fc5e0fa88e5e0e3afe37a492f8315002c9c15e9ba2ce707db7dfc797522b20312baa3071c8dafb76d1e32a3ccc00414ff5d7a29af6



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.