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 курсу

ФТІ группи ФБ-84

Григорян Володимир

Білецький Владислав

Перевірили:

Завадська Л.О.

Савчук М.М.

Чоний 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 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 $0 < k < n$.

Опис кроків протоколу:

1. Програма генерує 2 пари простих чисел. Для абонента A p, q та для B p_1, q_1 довжиною

   256 біт

2. Програма генерує ключові пари RSA для A і B, де (n,e)-Public, (d,p,q)-Private.
3. Абонент A формує повідомлення, використовуючи функцію SendKey(), де ще використовуються функції Encrypt(), Sign().
   k_1=1998031c378805936b19623728d6a15451d7f7cc1a55ea471b8aca363e7be18fe0cf0b04504fb95f49d265c247f3457cd4910f89c3550864fcc9e87c11f96d7
   S=29fb3aa431ae1d6fa7570b1380dba135c3ac23e069896cbdd771fba2123058dc7827170d19cb6b18a7b1c2654fb8cf14c24f75aa151cda565d6864cf4f0175c
   S_1=cd75a5a308a7e8d0444b68b1bdc3c3eb6afac63531507a8539fc44efe1d90ca50ec8cc37132c7a233fdca1127011c30fe027f93f1c23f27600b853c4047c835
4. Абонент B приймає повідомлення і за допомогою свого таємного ключа перевіряє підпис A.

   k=3403a3983c8c559547597a9e07f06ca882c899f5fa46a860481af6df4cdc9644806d8cb36dc9351f94f4511d5074f59654dc29b1033492423dcc6b9485ca2d4

   S=29fb3aa431ae1d6fa7570b1380dba135c3ac23e069896cbdd771fba2123058dc7827170d19cb6b18a7b1c2654fb8cf14c24f75aa151cda565d6864cf4f0175c

   $S^e \bmod(n)=k$, отриманий підпис правильний.

A(p,q) = 0xc61b64ef43b4c7d58de5f7638ef5eccb80db9d7345ff730369603c99d494b7d**,** 0x5a7f43976a265455786929f2507b4f77cf42c050bc5b2af4457388abb4300b83

B(p1,q1) = 0x4ee4655cf2264c6a180013b13fb2271a01202fc485f8921614f0e7588d6a198b**,** 0xc8a175b4354f160f1e9b7fe3e91ddf74e3edad85ac0a1fe1c6a80bd4152b67b1

A(n,e,d):

n=46081d6188384de8a5b52767f54e4b78ae2f9d64526be0b25990d2ccc1b057232183c824713f22551f883956e99483ea481164ef101c8737389753eae2ffff7 **– Open Key**

e=2231e5534de2779168e42b0e6370df4846cd6114bb098ca2386769074194785366073 22e3ba7fe8ac600d9f75534ff795e0b8929be056726a08cebd9aeaba1d **– Open Key**

d=4283173ecca3caffafbe1381b98acf432dfb937bd5e93b84bda9c62bf33e1c0695350c2c8f5453fe6601eb9bd583d402f84558b06f868a67365189f0fae83bd – **Secret Key**

B(n1,e1,d1):

n1=3dd4311a47fa9e1284aaa3789a1c4963c5d20a5daf1f3e6e6a3f0b3ea9a62c8061fa47a21ca7f31ac6186063c0b2ef97a0920d3384c4b418774c2a2627fb961b **– Open Key**

e1=40f4858e3a70b6d24262bacd1db104e3eadcc0d8370c3c9bb57ee67ddddb23a2b51c0bde7ce4de91de7e2e8304ba3be3449e3bdd85c72f4a6c1a0901ceab5f **– Open Key**

d1=16d40dd1076fc47b4a21e99bc190441e94c1d8d8fc3ae21bc41725d8a7ec5ad206067f36b618713cdd9c0b0b87242ca77c20c0e033ab0744a4bf5a7bdedb1a9f **– Open Key**

Open Text =
3403a3983c8c559547597a9e07f06ca882c899f5fa46a860481af6df4cdc9644806d8cb36dc9351f94f4511d5074f59654dc29b1033492423dcc6b9485ca2d4

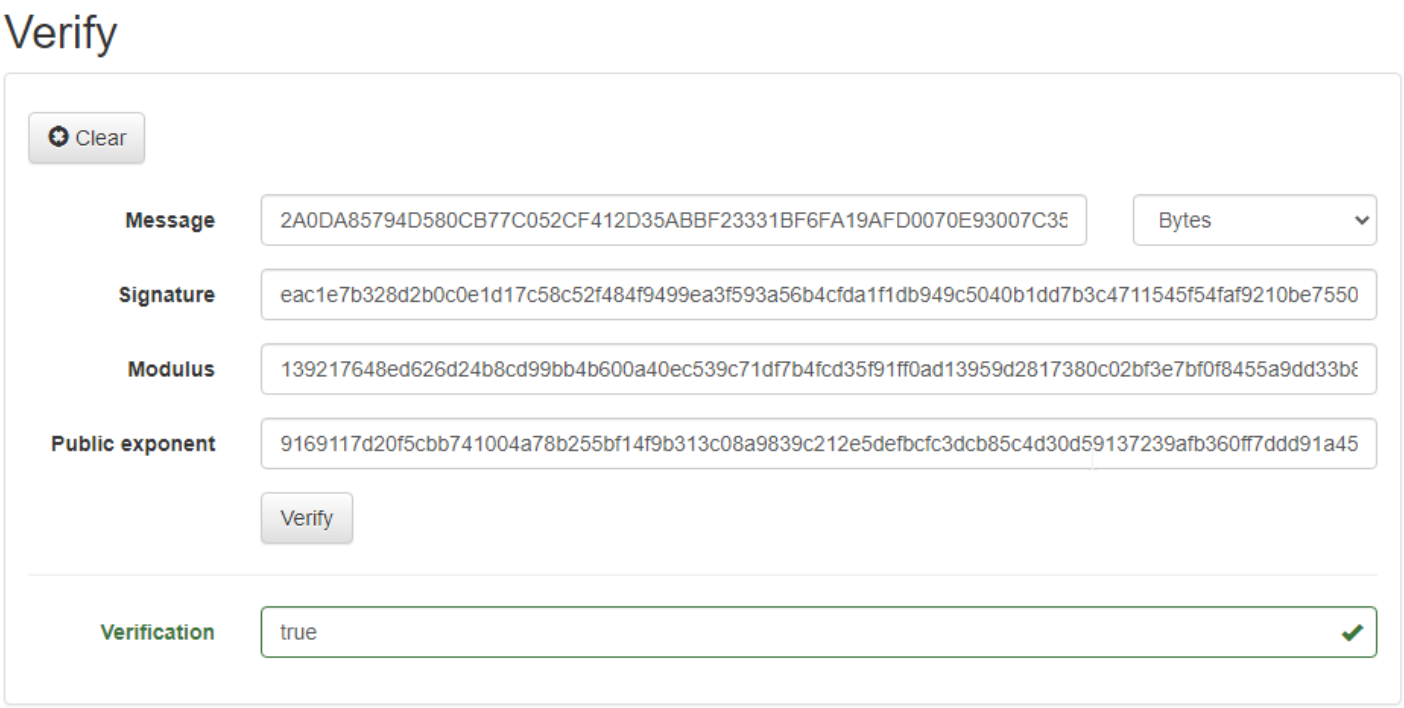
Encrypted Text =
1998031c378805936b19623728d6a15451d7f7cc1a55ea471b8aca363e7be18fe0cf0b04504fb95f49d265c247f3457cd4910f89c3550864fcc9e87c11f96d7

Цифровий підпис для A і B

A =
29fb3aa431ae1d6fa7570b1380dba135c3ac23e069896cbdd771fba2123058dc7827170d19cb6b18a7b1c2654fb8cf14c24f75aa151cda565d6864cf4f0175c

B =
cd75a5a308a7e8d0444b68b1bdc3c3eb6afac63531507a8539fc44efe1d90ca50ec8cc37132c7a233fdca1127011c30fe027f93f1c23f27600b853c4047c835

Обмін ключами з сайтом



Висновки:

Під час виконання лабораторної роботи №4 ми ознайомились з асиметричною криптосистемою RSA, генерацією ключів для цієї криптосистеми, а також з тестами перевірки чисел на простоту. У ході роботи реалізували функції шифрування, розшифрування, підпису, перевірки підпису для RSA. RSA-алгоритм з відкритим ключем, що часто використовується в криптографічних застосунках. Складність задачі цього алгоритму полягає у великих цілих простих числах.