

Міністерство освіти і науки України
Національний технічний університет України
"Київський політехнічний інститут імені Ігоря Сікорського"
Фізико-технічний інститут

Криптографія

Лабораторна номер №4

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

Виконав:

студент групи ФБ-81 Кудін Іван Антонович

Перевірив:

Чорний Олег Миколайович

Mema poботи: ознайомлення з тестами перевірки чисел на простоту і методами генерації ключів для асиметричної криптосистеми типу RSA; практичне ознайомлення з системою захисту інформації на основі криптосистеми RSA, організація з використанням цієї системи засекреченого зв'язку й електронного підпису, вивчення протоколу розсилання ключів.

Завдання:

- 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 попереднім тестом решета. Тести необхідно реалізовувати власноруч, використання готових реалізацій тестів не дозволяється.
- 2) За допомогою цієї функції згенерувати дві пари простих чисел p, q та p1, q1 довжини щонайменше 256 біт. При цьому пари чисел беруться так, щоб pq < p1q1; p і q прості числа для побудови ключів абонента A, p1 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),(e1,n1) та секретні d i d1.
- 4) Написати програму шифрування, розшифрування і створення повідомлення з цифровим підписом для абонентів А і В. 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 М і знайти криптограму для абонентів А и В, перевірити правильність розшифрування. Скласти для А і В 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.

Результат виконання:

RSA keyPair1 e= 65537

RSA keyPair1 n=

1005023880502734800836195990382337688890521114139325477440714081393495970330972888451095 1843287788198108545033571682631312727298865591092686108271544606261

RSA keyPair1 d=

5249090194199934311583098487083179998021521768752761305418039014910351949045299324676893 782305511807584860237093319462734218270357772171878727159566227073

RSA keyPair1 p=

115051471960843440636045322904291175410224620676606151450427830541068524632341

RSA keyPair1 q=

87354282685299681902588565564626868945355676465581129305734579598233535273121

RSA keyPair2 e1= 65537

RSA keyPair2 n1=

7085811362482970374253494691702052504183127445137684102900788640270511857476246472232507 781302170628684364463292871863489408877838693613904952901494877191

RSA keyPair2 d1=

6447475303705494183863445823737255244853326196437058701318075727989554505040976148119718 484009368938319071009037515884646723629353266915772314196737936521

RSA keyPair2 p1=

115714937640288389344438873007046910775953160422758957046598611903515441858763

RSA keyPair2 q1=

61235061842317481068402837417538809415036784265469292911686549398127326907957

Message 691766098505318584202167

Text 691766098505318584202167

ChipherText

1416851088497528638384017826503664836061034450577950987063511981791912519206196690367105 730413469802139819506875364602765511318512554257984739060126827636

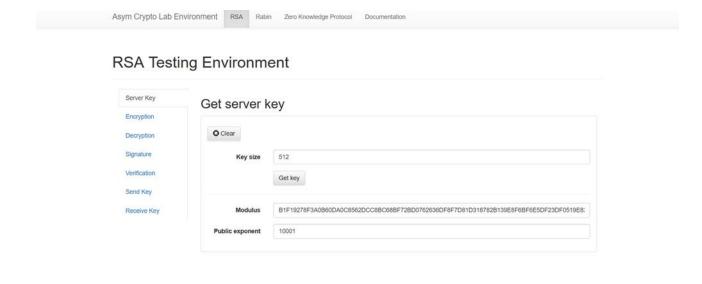
Text 691766098505318584202167

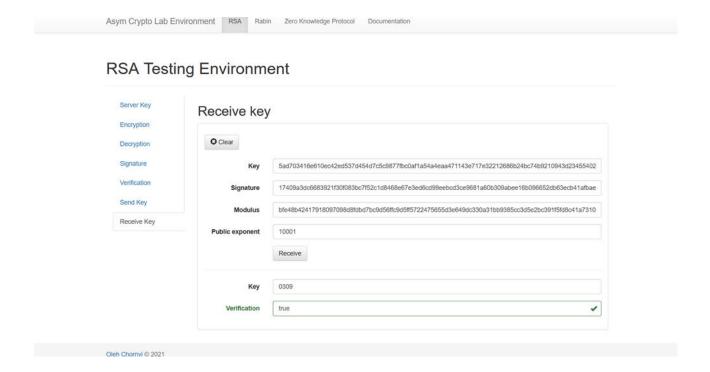
Digital signature is ... True

Tranfer key 207426983142543599565931936891661821545 True

Press any key to continue . . .

Oleh Chornyi © 2021





Висновки:

- 1. В ході виконання роботи отримано навички роботи із криптосистемою RSA, зокрема побудовані функції генерації ключів, направленого шифрування, виробки та верифікації цифрового підпису, транспортування ключів.
- 2. Сучасні вбудовані реалізації арифметики великих чисел Python3 забезпечують достатню ефективність реалізації за швидкодією
- 3. Швидкість реалізації криптографічних систем в системі програмування Руthon3 достатня для практичних застосувань.