



Міністерство освіти і науки України
НТУУ «Київський політехнічний інститут»
Фізико-технічний інститут

Лабораторні роботи № 4

з предмету «Криптографія»

на тему: «Вивчення криптосистеми RSA та алгоритму електронного підпису;
ознайомлення з методами генерації параметрів для асиметричних криптосистем»

Виконали:

Студенти 3 курсу ФТІ

Групи ФБ-84

Зеленін В.Ю.

Михайленко О.В.

Перевірив:

Чорний О. М.

Мета 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.

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 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 $1 \leq p, q$ довжини щонайменше 256 біт. При цьому пари чисел беруться так, щоб $pq \leq p_1q_1$; p і q – прості числа для побудови ключів абонента А, $1 \leq p \leq q_1$ – абонента В.
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 А і В – тобто, створити та зберегти для подальшого використання відкриті ключі (e, n) , $(,)$ і n і секретні d і d_1 .
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 M і знайти криптограму для абонентів А і 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 $0 < k < n$.

Хід роботи:

Під час виконання лабораторної роботи ми стикнулись з наступними проблемами:

- Необхідність використовувати функцію `row`, вбудовану у пайтон, оскільки ми при використанні звичайних операцій програма некоректно реагувала на великі числа ключів.

1. Робота виключно з локальними функціями

Примітка: числа, які було відкинуто під час тестів, додаються до файлу `log.txt`, де їх можна переглядати.

Таблиці значень елементів ключів для кожного з учасників (як приклад):

A		B	
Open Key			
n	52411688293442316707227815200226897051 14387902571992526187948318920290293191 96038433631036378141911225677046771470 79155946694887859203386624618678923988 9		1144232297228601879356868369810509886624 6484261620545723218527012140908861975826 0416578138742028965433828378125500981626 40092167858503288370078515047607347
e	52125514618417016069366477217654015518 44233475265781331117725550045307025580 63172176357311958050253181213617559637 86185763351033837540646464763132379304 1		2121095282572591147787725399989764776433 7771451438172431878136866781441185036466 2411580663707041669030127425769510845587 390987158954082529043500132625137
Secret Key			
p	14967568988040225479967761104497604129 96147451549396790647657154073603844558 9		1055367132328984143425277932167911862911 68911776980645283478423770722953165579
q	35016834287065361753195441098960275281 19549978150313319706176940019860853870 1		1084203081730913955746138435571467068429 71801986983869070489609587396281961593
d	12262787970074932551657829036224960114 73250394644484293953280683255904365053 71520780688139843255658834771874123777 78438964427705480930406587005907363761		4381596033746114716132944628514471 9375976861590000274541986278810470763653 1468229883085947460397161878873206064576 4623599459696068504949436333451426137409

Приклад шифрування/дешифрування

A		B	
P	30		330
T			
C	3207853946116805334953165234176750650160 9196445937568554113675016671893599694036 8110973761666225723007566962372124872653 904619374602377726886218559896655		6575896556963011793638303832247899573724 3649249749463484900404045395831216159930 3659219862361293635978050940593071555965 7745646157666620980204368551885123
T			

2. Робота з допомогою сайту

За допомогою сайту <http://asymcryptwebservice.appspot.com> було перевірено створені функції.

Я згенерував 512-бітний ключ на сайті та за його допомогою передав сайту таємне повідомлення

Me		Asymcryptwebservice
Open Key		
n	a01d4783218ce95acf552ee6de3ec197c0df6e193642ebc6d62fca44fa85da04e59730408047397a57f8a4527f161a76679d929ca2008c8ce803a446f7ffc51	8EB8F87CE1564A1B1DCABCAC8A92DCC298A315F122C3148E1E7D73184BA09CC7A1AC8775CC7FE3735CC80C3271E385E17E2D72B3CB1428669AAE03870B803DF7
e	9f3d792a1e5357fde7909a5f17f63bf884caa5126d871bbca0f780c3434cad40e245408f17c3a0bbd07ad83e9aa305da31cea29708954de12dd96aa9c00691	0x10001
Secret Key		

Sent Data	
k 1	42533ffd23b441185164b53a595079f5482e0b600c0c853b9080e5d0b5ef85d62f3b7598db112bd1c0aa277e8be69bdc646e7a640ee7af6bb53eea018eb22930
s 1	5625725db15c63ccd0f17404f1db476e3f1da7237b8cf409e02d5bee8f7c1e8c5b475151b7634512e9daeab7880cee19d7538e36277f7a36715634a301c9ead4

Asym Crypto Lab Environment **RSA** Rabin Zero Knowledge Protocol Documentation

RSA Testing Environment

Server Key
Encryption
Decryption
Signature
Verification
Send Key
Receive Key

Receive key

Clear

Key 42533ffd23b441185164b53a595079f5482e0b600c0c853b9080e5d0b5ef85d62f3b7598db112bd1c0aa277e8be69b

Signature 5625725db15c63ccd0f17404f1db476e3f1da7237b8cf409e02d5bee8f7c1e8c5b475151b7634512e9daeab7880cee

Modulus a01d4783218ce95ac552ee6de3ec197c0d95e193642ebc6d62fca44fa85da04e99730408047397a578a45271161a7

Public exponent 9f3d792a1e53579de7909a5f1763bfb84caa5126d871bbca0f780c3434cad40e24540817c3a0ebd07ad83e9aa305

Receive

Key 1E

Verification true ✓

Висновки:

В результаті виконання лабораторної роботи ми ознайомились 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.