Міністерство освіти і науки України Національний технічний університет України “Київський політехнічний інститут ім. Ігоря Сікорського” Фізико-технічний інститут

Лабораторна робота №4 з предмету «Криптографія»

« Вивчення криптосистеми RSA та алгоритму електронного підпису; ознайомлення з методами генерації параметрів для симетричних криптосистем»

Варіант 10

Виконав студент 3 курсу

Нечаєв Олексій ФБ-02

Київ - 2023

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 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.

Результати виконання:

```
Числа для абонента A:
p: 767831709558001550688721924463020959927419944189080671991621988130394074039 63
q: 833743907920638316314011564775992906646309095079425222752313648183618983622 51

Числа для абонента B:
p: 885467890439793986607384680988152384376249871517685729552673750600047156910567
q: 857980859124496912487457490004581479995179824561358334957134190373955128927 11
```

```
e:  473152707243087083915617835274123487479335760521019704661571711467658588322523681802640856069492586091916528081552519012509188745010190261098984425386684 9
n:  640175010152272748283261246692745340644001426429100182031250753848846220722217682855337338885138544207886601240676679193524338168641776965675177326700071 3
d :  469554051711368944456523597334616678026870971082968410637517721077630633244175469479165777771442190449574986820391056782521797101557286367508208608029664 9

e1:  424805569678202015519267624706827442073462346960796431224918207769779013151380912253155651806083452064136086332550511561871874777491449780182492041968516 7
n1:  759714501366690350878975032939878427004316473239157529087908550352824312035180259288725848096568994368051247600528919906254261941383263261774866809317713 7
d1:  206266340634320895572179866474674650656196690917273970093961613180193487333090079911038635603600354845383451965946478545641469508492599343160699837567688 3

Відкриті ключі [e,n] для абонента A:
e =  473152707243087083915617835274123487479335760521019704661571711467658588322523681802640856069492586091916528081552519012509188745010190261098984425386684 9
n =  640175010152272748283261246692745340644001426429100182031250753848846220722217682855337338885138544207886601240676679193524338168641776965675177326700071 3
Секретний ключ для абонента A:
d =  469554051711368944456523597334616678026870971082968410637517721077630633244175469479165777771442190449574986820391056782521797101557286367508208608029664 9
p =  767831709558001550688721924463020959927419944189080671991621988130394074039 63
q =  833743907920638316314011564775992906646309095079425222752313648183618983622 51

Відкриті ключі [e,n] для абонента B:
e1 =  424805569678202015519267624706827442073462346960796431224918207769779013151380912253155651806083452064136086332550511561871874777491449780182492041968516 7
n1 =  759714501366690350878975032939878427004316473239157529087908550352824312035180259288725848096568994368051247600528919906254261941383263261774866809317713 7
Секретний ключ для абонента B:
d1 =  206266340634320895572179866474674650656196690917273970093961613180193487333090079911038635603600354845383451965946478545641469508492599343160699837567688 3
p1 =  885467890439793986607384680988152384376249871517685729552673750600047156910567
q1 =  857980859124496912487457490004581479995179824561358334957134190373955128927 11
```

```
Початковий k =  49370615552818110693258255989533005333063613703339281812445132983603204499743420641782346127981577436913794794377400273695682588726356180810766116174443 20

Повідомлення:  44588061126867135071633027441819389665093770597627704789488684735062308704173124877041686040410607793038581033281308196372735050052617443212970170663395 71

Розшифрований k:  49370615552818110693258255989533005333063613703339281812445132983603204499743420641782346127981577436913794794377400273695682588726356180810766116174443 20

Ключ отримано

Шифрування:  52807912385778031246203707543750477677186868147901697359135031677041559747137973992362365682526334486848267213552356520725229762380016582810037276698384 4

Розшифрування: 44588061126867135071633027441819389665093770597627704789488684735062308704173124877041686040410607793038581033281308196372735050052617443212970170663395 71

Функція Ейлера: 64017501015227274828326124669274534064400142642910018203125075384884622072220166709916255248646851687299421110201094190313165310969433760931203719612345 00

Перевірка:  True
```

Висновок:

В ході виконання лабораторної роботи було здійснено ознайомлення з тестами перевірки чисел на простоту та методами генерації ключів для асиметричної криптосистеми типу RSA. Вивчено протокол розсилання ключів для асиметричної криптосистеми RSA