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 3

Виконали:
Студенти 3 курсу,
ФТІ, груп ФБ-02, ФБ-05
Кодак Єгор,
Нікітський Іван

Київ - 2022

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.

Хід роботи

Було реалізовано такі методи:
generate_key_pair_rsa - генерує пару відкритий та закритий ключ
encrypt, decrypt - зашифровує та розшифровує протокол
signature - обчислює сигнатуру
verify - верифікація сигнатури
send_key - шифрує ключ і сигнатуру та надсилає їх
receive_key - отримує ключ і сигнатуру, розшифровує та валідує їх

Результат:

```
client_a info:

[+] public key:
        e: 24932033475508689213762795956443295993014227848816802638904568314250507626043421369361692124142410634406621556812761310171303168960350988994252728
88763661
        n: 63188230968607891169737355298469712534093437123339082618540606062140634612338268536038799031485556841358741335425158597975737002531066864457812723
96324553

[+] secret key:
        p: 9656940575926373992547026795045544391108521384555087754186766452748925202820l
        q: 6543297069273552088649272389796406459121320237623895547669301726379871164995 3
        d: 4580430464994954893944521117958921009793054393042484719059193674803404897929064775501516640191975370699553868432540742511873986276863233884706505913254341
```

```
client_b info:

[+] public key:
        e: 33456518596205036432014533750646325053068854200229558200034729954171125730968420993509421649260223910438860755673475559940182802724938576118305236
9872133
        n: 80545023801846758450147142112763358741085365577233432429685654190684471159463545393270506992419431074650805031331150161748834352644438228537778472
81558123

[+] secret key:
        p: 870458123446835060965592205797363341429031486536177258513500397202453997107 83
        q: 9253176187603952429428171429236597003999957862289763527620646981132868016098 1
        d: 1599033924087546141540797446397477741898977884834198876023479026514300123320316856370474482549637221727209195177235987852353418238998294806611724217407157

Message: 22832213371488666777
Started k: 1988915344975632760187763439452233652703952582061722768592441095261098416658153418954904940486169902972162801087719448396421160028726702830505480058257416
Encrypted key: 5360249258498076377586636910712894102234941954300763731920730772310910262558761208919856377997244867025178606385159235795218181154629002703622191371298615
Encrypted signature: 5203175370681190308990237111716538763488121170553001077495527237473206508020207753613669519516947450337055411926873730589465197020334697179380021929508 43

Encrypted message: 2035106119850184562665383677613212586044511815850185486465234261050160919227490886217217219592908243416459235767420178572253347581696409514299635117131253
19889153449756327601877634394522336527039525820617227685924410952610984166581534189549049404861699029721628010877194483964211600287267028305054800582574 16
Decrypted message: 22832213371488666777
```

Перевірка:

Висновки
В ході виконання лабораторної роботи було здійснено ознайомлення з тестами перевірки чисел на простоту та методами генерації ключів для асиметричної криптосистеми типу RSA. Проведено практичне ознайомлення з системою захисту інформації на основі криптосхеми RSA, організовано з використанням цієї системи засекреченого зв'язку та електронного підпису. Вивчено протокол розсилання ключів для асиметричної криптосистеми RSA.