МІНІСТЕРСТВО ОСВІТИ І НАУКИ, МОЛОДІ ТА СПОРТУ УКРАЇНИ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

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 гр. ФБ-14

Цибулено-Сігов І. М.

Татаренко А. О.

Перевірила

Селюх П. В.

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 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 1 1 pq p q ; p і q – прості числа для побудови ключів абонента A, 1 p і 1 q – 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 1 e n та секретні d і 1 d.
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.

При генерації підходящих під умови криптосистеми випадкових простих чисел зіткнулися з проблемою, коли такі 4 числа генерувалися дуже довго. Це було пов'язано з тим, що генерація останнього числа - q2 залежала від попередніх чисел, і це число могло мати дуже малий інтервал, тільки попавши в який, воно б задовільнило рівняння p*q < p1*q1. Вирішили цю проблему обмеженням кількості перегенерацій числа q2 (5), якщо за таку кількість q2 не задовольнятиме рівняння, всі числа будуть перегенеровані.

Через умову криптосистеми щодо p q та p1 q1, кандидати можуть відкидатися:

```
Не пройшов q2 - 80134130270402594791740897332352202994236215239971943436339165285193902610423
Не пройшов q2 - 42039859030633443063851639058323548946666709122867443708426234008238286801339
Не пройшов q2 - 91507773162476816070272939056889164418829096231538599195577484375748357128611
Не пройшов q2 - 79162912736260702962844790083935057847586757596308712920721035291770933734723
Не пройшов q2 - 88996286057809281532230717002127299053467662220811043324650508257729375222243
Не пройшли:
 p1 - 444157512891392109824114915705799721124537855333255994866557483392565898397 59
 q1 - 96085064245935949831893874137027458676171358728043968112523223926612439050987
 p2 - 25184690510580494655624164489673533868199575447636937246746625318682471398283
```

В результаті, отримали пари ключів:

```
p = 54096515714858767869060490161884628381484798405909961334005818854896734155263
q = 10593233340232226030649273709567208487438093821681448949948513984318625683232 3
Public key A: [474521387424883950821097498409302742798102228392876103008614080115664329274736007918821585
Private key A: [540678243594083348028824222169980460367715790893579843144514543694285473497223352241785837
p = 95504194371631507434182771786847392898508845554858822865728493648973593947671
q = 62035423351955603901592320145511541745286458219792982762070983430161438387459
Public key B: [165200825834872442923715706567075555450613116468427245728784683867386498642651072791465166
Private key A: [192841940005690899700238199983245412671696871345556664582557215523418025737663870690464160
```

Моделюємо надсилання ключа k:

В надсилає ключ A. Для цього він спочатку зашифровує повідомлення публічним ключем A:

```
print("B ---<message>---> A")
message = randint( a: 2,  b: 10000)
print(f"Message: {message}")
c = Encrypt(message, pub_key_a)
```

Потім підписує криптограму:

```
signed_message = Sign(c, pr_key_b)
print(f"Signed message: {signed_message}")
```

A отримує підписане повідомлення та перевіряє підпис, у разі підтвердження підпису A розшифровує повідомлення:

```python
if Verify(signed_message, pub_key_b):
    print('Message from B is verified.')
    m = Decrypt(signed_message[0], pr_key_a)
    print(f"Decrypted message: {m}")
```

Приклад:

```
A ---<message>---> B
Message: 8522
Cryptogram: 14856773174318712360312692449035394367802297517310144564216150866066495648148193
Signed message: (14856773174318712360312692449035394367802297517310144564216150866066495648
14856773174318712360312692449035394367802297517310144564216150866066495648148193967675609819
Message from B is verified.
Decrypted message: 8522
B ---<message>---> A
Message: 123
Cryptogram: 542188424562664611687957698060737801016245185560012707406946880574574267426548157
Signed message: (542188424562664611687957698060737801016245185560012707406946880574574267426
542188424562664611687957698060737801016245185560012707406946880574574267426548157056576670825
Message from B is verified.
Decrypted message: 123
```

Висновки:

В ході виконання лабораторної роботи ми вивчили й використали на практиці методи знаходження простих великих чисел, генерації ключів для асиметричної криптосистеми типу RSA та системи електронних підписів.

В результаті виконання лабораторної отримали імітацію обміну повідомлення з шифруванням, підписом та його підтвердженням.