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

КРИПТОГРАФІЯ КОМП’ЮТЕРНИЙ ПРАКТИКУМ No4

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

Виконали: Бойко Т. Я.

Хандрос А. В.

Група: ФБ-02

Київ – 2022

Мета:

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.

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.

Для класу RSA розроблені методи mr_test(), isprime(), prime_gen()

2. За допомогою цієї функції згенерувати дві пари простих чиселqp, i11 , qp довжини щонайменше 256 біт. При цьому пари чисел беруться так, щоб11qppq ≤ ; p і q – прості числа для побудови ключів абонента A,1p i1q – абонента B

Це завдання виконують функції smallbig_gen(), pairs_gen()

3. Написати функцію генерації ключових пар для RSA. Після генерування функція повинна повертати та/або зберігати секретний ключ),,(qpd та відкритий ключ),(en . За допомогою цієї функції побудувати схеми RSA для абонентів A і B – тобто, створити та зберегти для подальшого використання відкриті ключі),(ne ,),(11 ne та секретні d i1d .

Ключі зберігаються у відповідному об’єкті класу ABconversation, генеруються методом RSA.keys_gen()

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.

Для класу ABconversation реалізовані методи encrypt(), decrypt(), sign(), verify() а також модифікацій цих методів для роботи із хексовими значеннями.

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 $0 < k < n$.

Продемонструємо роботу протоколу

Генерується випадкове повідомлення

```
Message: 437926966
```

Шифруємо повідомлення ключем B, A підписує

```
msg_enc = A.encrypt(msg, B.e, B.n)
sign = A.sign(msg)
print(B.e, "\n", B.n, "\n")
```

343070937658085375591515992522814183142973683715922229536553989493072988265878016322619524944781926136597444278162715788818813546893009574297721197650910186979855661836407512779310689818134684415478946508199045078928348168682295964167089966953404909823582412106912066783585360361832059720798276746483978496731541615085397006782975485282690827358322575295779470102967543347769663019097276097840119662479085463175059402874004351810012849666357042063318102073021954116154161508539700678297548528269082735832257529577947010296754334776966301909727609784011966247908546317505940287400435181001284966635704206331810207302195184758167011244058871028320882186767118074058589719847546744845130799184885674037550766431095517042884422631487642164619689503420797543631295499556955015518475816701124405887102832088218676711807405858971984754674484513079918488567403755076643109551704288442263148764216461968950342079754363129549955695501551
```

Тепер розшифровуємо

```
msg_dec = B.decrypt(msg_enc)
s_verify = B.verify(msg_dec, sign, A.e, A.n)
```

Decrypted msg: 437926966

Verify: True

Висновок

Виконуючи лабораторну роботу був реалізований протокол шифрування RSA (перевірка чисел на простоту, генерація та обмін ключів, робота цифрового підпису. Перевірено роботу програми на випадково згенерованому повідомленні.
```