



Міністерство освіти і науки 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

Виконав:
Студент III курсу ФТІ
Групи ФБ-06
Сулима Олексій
Перевірила:
Селюх П.В.

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.

1. Написати функцію пошуку випадкових 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 і довжини щонайменше 256 біт. При цьому пари чисел беруться так, щоб p і q – прості числа для побудови ключів абонента A , і – абонента B . $q \mid p-1$, $p \mid q-1$, $q \mid p+1$, $p \mid q+1$
3. Написати функцію генерації ключових пар для RSA. Після генерування функція повинна повертати та/або зберігати секретний ключ та відкритий ключ. За допомогою цієї функції побудувати схеми RSA для абонентів A і B – тобто, створити та зберегти для подальшого використання відкриті ключі, та секретні і.), (qpd), (en), (ne), (1 1 n e d 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.
5. За допомогою датчика випадкових чисел вибрати відкрите повідомлення M і знайти криптограму для абонентів A і 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. $nk \square \square 0$

Перевірка за допомогою сайта

a Open key is:

554960030893671090694285255452065160258954261216541142677440348036026435259222609798620
791814584566055435461347012295152571798196180306351939380212805499394562978650679831705
751239271259227601250867304345577201131310037126305179745718922271327585514264236637364
92045036945639382719006640957187855605213827

a private key is:

189285219628189505976094053818540069226983589848795219028596922949278362432036358533630
991078664567376985241682125751525937778065831885502892986850877589919884260034235804819
701800905325814101104052315920617765341049566122607632691021276630855825009735107820086
5109718215073396747171411744624255561009012583

b Open key is:

830704862698583939442599760075045852399975833580339768400273305373034349720700764847565

065103748057998657771985698845227540975493798671248295547161505539635001483981600248865
362332311060709840710923743127626535576499632822819386074367076176375440614941353921961
50522700709024408493542182360292989143715843

b private key is:

845796577895007639531659127583414401182626166566253166452007875936372595909345404838166
633774652741577719072886752843247038654032193819803759864149261865240084687141131217779
635466134215824204225314809727095771507279349498752327594013167852857590866087429878335
4939513148588519091005259142334460005111182677

Sign:

306710679525634120506719433876111289919567153557965437705901072205060095573454461999878
444570731832185531056487638907063853737986817204894588687406501289

Server Key

Encryption

Decryption

Signature

Verification

Send Key

Receive Key

Get server key

✖ Clear

Key size

305

Get key

Modulus

1DE0C03DC851BF5BCC227ADE52FC439A526C2F3E34F24AA8A

Public exponent

10001

leh Chorny © 2023

7:04 PM

1/26/2023

Не захище | asynccryptwebser...

Encryption

Key

Encryption

Decryption

Signature

Verification

Send Key

Receive Key

Clear

Modulus: 1DE0C03DC851BF5BCC227ADE52FC439A526C2F3E34F24AA8A

Public exponent: 10001

Message: 312

Bytes

Encrypt

Ciphertext: 01D4AA5379B1DE1DC579F587768D0278D7F51918AA4406CBA8

Chorny © 2023

100%

7:05 PM 1/26/2023

Decryption

Clear

Ciphertext: 01D4AA5379B1DE1DC579F587768D0278D7F5

Bytes

Decrypt

Message: 0312

цієї системи засекречений зв'язок й

Sign

Message

0312

Bytes ▾

Signature

016DEA43E22099C3C2B684FAC1B3D14F922131445A96691590E

3

Verify

Message

312

Bytes ▾

Signature

016DEA43E22099C3C2B684FAC1B3D14F922131445A96691590E

Modulus

1DE0C03DC851BF5BCC227ADE52FC439A526C2F3E34F24AA8A

Public exponent


10001

Verification

true

✓

Send key

 Clear


Modulus

Public exponent

Key

Signature

Receive key

 Clear


Key

Signature

Modulus

Public exponent

Key

Verification 

Висновок:

Ознайомилися з тестами перевірки чисел на простоту і методами генерації ключів для асиметричної криптосистеми типу RSA; практично ознайомилися з системою захисту інформації на

основі криптосхеми RSA, організували з використанням цієї системи засекречений зв'язок й електронний підпис, вивчили протокол розсилання ключів.