МІНІСТЕРСТВО ОСВІТИ І НАУКИ, МОЛОДІ ТА СПОРТУ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

ФІЗИКО- ТЕХНІЧНИЙ ІНСТИТУТ

Кафедра інформаційної безпеки

КОМП’ЮТЕРНИЙ ПРАКТИКУМ №4

з дисципліни

Криптографія

З теми: « Вивчення криптосистеми RSA та алгоритму електронного підпису; ознайомлення з методами генерації параметрів для асиметричних криптосистем»

Варіант 18

Виконав студент групи ФБ-91

Свищо Максим Іванович

Мета 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.

Завдання:

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:

Кандидати на ключ, що не підійшли:

Абонент А:

P =
3906603036928247948694095696842510335337518968242405050872411759300788158592 01

Q =
2819049808656521948482712233415749114204190729565810211815672193682595875679 89

N =
1101290854374956492729509868144553951030282765540084696406811759307350152293 5889264717199766998024858473926556753994417985400853199141501312694695103871 6789

E =
28794724804082479379767795310126883569224445670380202118809714094454721840834369695116486051118408747903524547241269830393060336447693579253490190489922699

D =
39346154362508720079750405453439867159979795407436555867912636616650791353806133275032197379085146167678735927783148474521171481367118861909279529295825699

Абонент B:

P =
32647250271496030923065694278067640208595728082341298407218400424764704092325 3

Q =
41245298099844199175790252134436033181050046815520648115357717879928035287555 7

N =
13465455695880732598881817299908420678672563430216930461976070859718988756726542283733334241658797702169103320466090467974936130685474256798716303469662692121

E =
65096832897326821394456468728698464843512796960896914820578677050181719281694177094910565848323538508239041413344461700659580604558067701246711682668917 1

D =
80421462714851702901474725204549493192156305179676581533344217060620769716117023969984867499050400015860874704101045639833894849540061665439717517773571675

Відкритий текст:

309891421540156697273082459497171885042138087559440383141185535298079385976 43

ШТ по ключу Б:

52211167354374419418385820097839803795289845446769523985997620110529854079217815009357864409712107060834212432528847771226935182643181985815376067559108144

Підпис Б:

63580611191303342501885885998182737858873230710075636886607732210844721949155161203345179213489299670424448064245855017310791323961068691448973570436967922

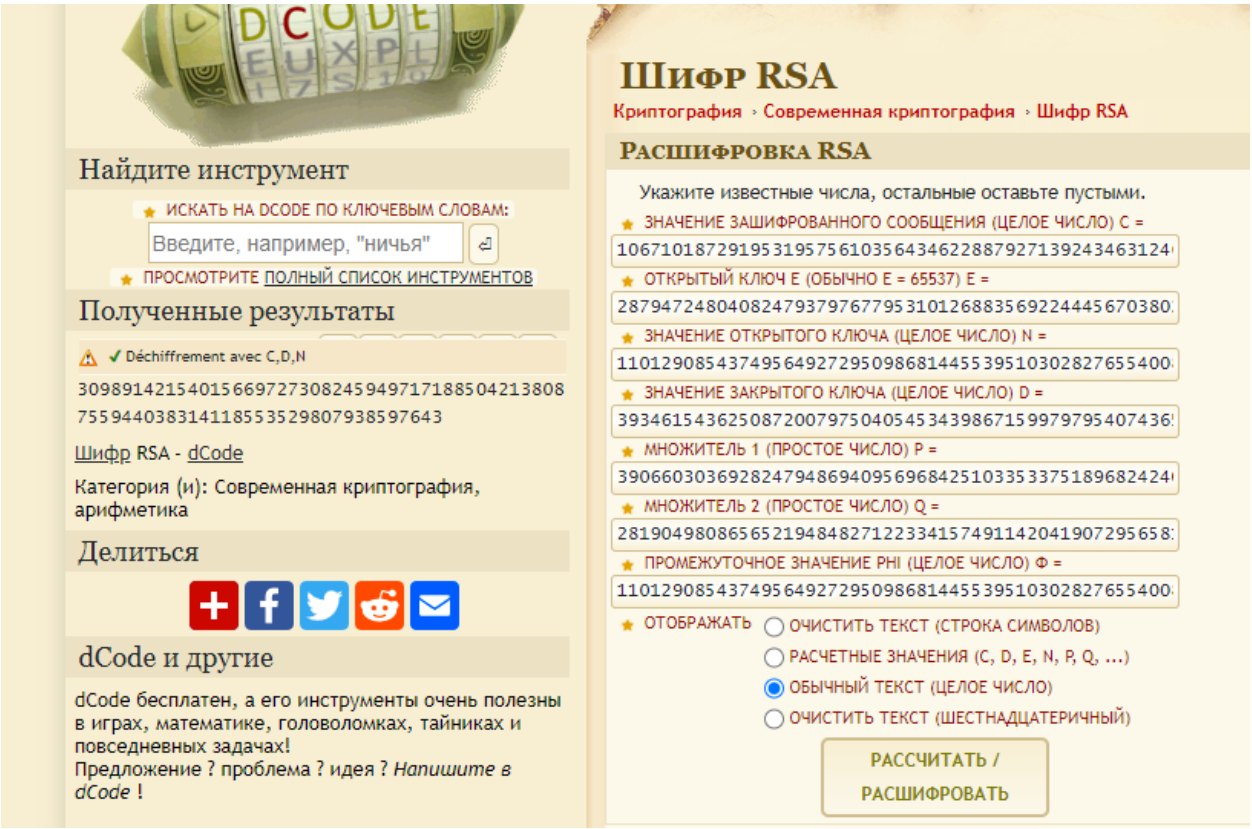
ШТ по ключу А:

10671018729195319575610356434622887927139243463124680004392383365147541628415960768202519643590032875753154407087560579534354799876334479314933664446546953536

Підпис A:

124649326120788666097040622058048184957487434833888866157833161905113140346575462215958984764855755498110019150459196856082513619110942744142692889958648728

Перевірка роботи на сайті https://www.dcode.fr/chiffre-rsa



Результат роботи програми:

```
 309891421540156697273082459497171885042138087559440383141185535298079385976437
Message was generated
Message was received
309891421540156697273082459497171885042138087559440383141185535298079385976437 Verified from A to B
Message was generated
Message was received
309891421540156697273082459497171885042138087559440383141185535298079385976437 Verified from B to A
```

Key =
270765368342928357813035903803294328026817058289467921444477069557514894321 54

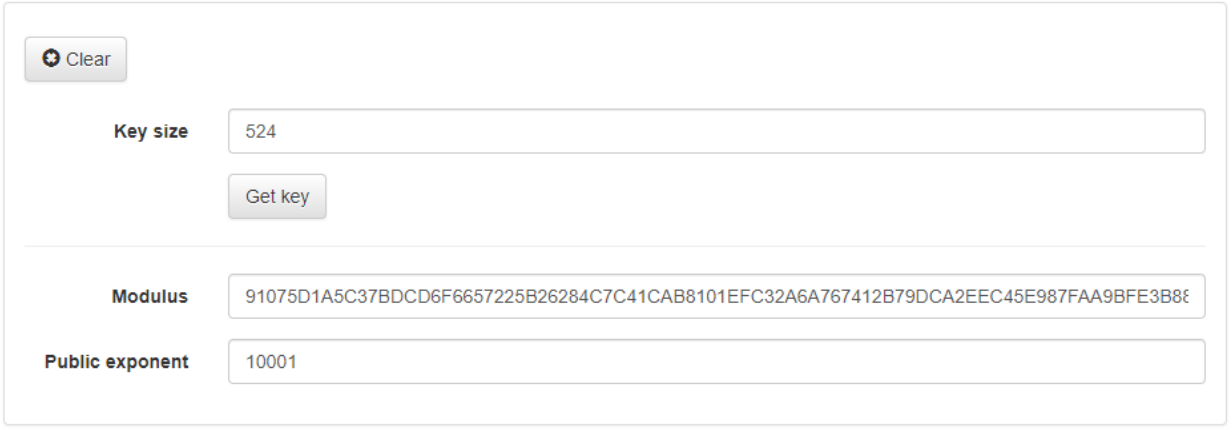
N, e =
(146225755592901409881041475380606689291864418604705072961172629753497120087902078556926046454692372312086009627277109555508277867505144846909612674681187 13,
114766080980218954015996265107140374811402002326402289521644377306920558261156344079860920254479441467119558682191815854756885290017563737569648604326666 7)

Message was generated

K, s =
(29686929837638623639783111494219697689317427567719155624875492592842408649665

2840280976173155854676595774838196258558003693480330341814032207365667062460005743,
5978825210405189445019246495713429214579441610555569220429094418933121986709918118259954002846383244116792032226917546617347615113127656285705347679378548520)

Get server key



Receive key