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

Виконали:
студенти групи ФБ-32
Гереновська Мирослава
Клименко Іван

2025

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 p_1, q_1 довжини щонайменше 256 біт. При цьому пари чисел беруться так, щоб $pq \leq p_1 q_1$; p і q – прості числа для побудови ключів абонента A, p_1 і q_1 – абонента В.

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 А і В – тобто, створити та зберегти для подальшого використання відкриті ключі (e, n), (e_1, n_1) та секретні d і d_1.

4. Написати програму шифрування, розшифрування і створення повідомлення з цифровим підписом для абонентів А і В. 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 М і знайти криптограму для абонентів А и В, перевірити правильність розшифрування. Скласти для А і В 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.

Варіант: 4

Хід роботи:

Ми реалізували:

1. Швидке піднесення до степеня за модулем: ми використали алгоритм піднесення до квадрату та множення для ефективного обчислення $a^b (mod\ m)$.
2. Розширений алгоритм Евкліда: ми застосували цей алгоритм для знаходження найбільшого спільного дільника та обчислення мультиплікативного оберненого елемента
3. Тест Міллера-Рабіна: ми реалізували ймовірнісний тест на простоту з кількістю раундів k=40, що забезпечує малу ймовірність помилки.
4. Генерація простих чисел: ми застосували комбінований підхід. Спочатку згенероване випадкове непарне число ми перевіряємо на подільність малими простими числами (3, 5, 7), що дозволяє швидко відсіяти очевидні складені числа. Тільки після цього ми застосовуємо ресурсномісткий тест Міллера-Рабіна.

Опис труднощів що виникли та шлях їх розв'язання:

Під час реалізації протоколу конфіденційного розсилання ключів ми виявили проблему коректності відновлення повідомлення. Ми з'ясували, що якщо модуль абонента-отримувача (n_B) менший за модуль відправника (n_A), то при операціях шифрування та підпису може відбутися втрата інформації (результат обчислення перевищить модуль n_B і буде взятий за модулем, що унеможливлює коректне розшифрування).

Що ми зробили: У програмний код ми додали перевірку умови $n_B \geq n_A$. Ми реалізували цикл, який автоматично перегенеровує ключову пару для абонента В доти, доки його модуль не стане достатньо великим для коректної роботи протоколу.

Результати генерації параметрів криптосистеми:

<u>Абонент А (генерація ключів):</u>

Під час пошуку простих чисел ми відсіяли ряд складених чисел. Приклади кандидатів, що не пройшли нашу перевірку:

```
Генерація параметрів криптосистеми для Абонента А

Генерація ключової пари RSA довжиною (256 біт)
Кандидат 104500202082711807500631340159917262671295679809581805070869622737985298439421 відсіяний тестом Міллера-Рабіна
Кандидат 677661841923408912956306198984361722294170523295543093834673383438823430817 87 відсіяний перевіркою пробним діленням
Кандидат 984493371841559760586472636010788804194745179331575531848335240243041315582 07 відсіяний тестом Міллера-Рабіна
Кандидат 108893592479359316377793879545901602538934817042342548830944960453116828974075 відсіяний перевіркою пробним діленням
```

У результаті ми отримали наступні параметри:

Просте число p:

```
A p: 651083955209222227710239522048874551057604809533181287320576666624224364755 81
```

Просте число q:

```
A q: 90865760060658208458973058430455714316419303730332567300658215057705804308427
```

Модуль $n_A = p \cdot q$ та відкрита експонента e:

```
Відкритий ключ A (e, n): (65537, 59161238453385523009157653577053380558055560020051825140879014230710394297671063007093625364863877791188010590593852906853805926875077880682927869780210 87)
```

Секретна експонента d_A:

```
Секретний ключ A (d, n): (358639675681607861762183868496079124188014303986547898289818324972442018257184749872064884878946798793620879391171924917182337928772011448342684527780 7833, 59161238453385523009157653577053380558055560020051825140879014230710394297671063007093625364863877791188010590593852906853805926875077880682927869780210 87)
```

<u>Абонент B (генерація ключів):</u>

Ми забезпечили виконання умови $n_B \geq n_A$.

```
Генерація ключовоі пари RSA довжиною (256 біт)
Кандидат 10348117716011637899979371245276373145836022128983793054336781151121748651937 7 відсіяний перевіркою пробним діленням
Кандидат 69718799791128422753946108065289405964259202111382333951973055076532769386795 відсіяний перевіркою пробним діленням
Кандидат 90487626125793934706695521949676097232657406127846560660890049077499856555167 відсіяний тестом Міллера-Рабіна
Кандидат 11510157321148209622933213629674653963433563302783739330240232572832035192532 9 відсіяний тестом Міллера-Рабіна
```

Просте число p:

```
B p: 83962147924664787170849855646199612810473846341963406086649619760350705551593
```

Просте число q:

```
B q: 10865410557363831734878986722616554606564000930592322672465159408547526459100 1
```

Модуль $n_B = p \cdot q$ та відкрита експонента e:

```
Відкритий ключ B (e, n): (65537, 91228320847959651315590708831340204049773457952419290808011790142695905004364637713911427940024014510316500412383083724307748611361219068377046272490145 93)
```

Секретна експонента d_B:

```
Секретний ключ B (d, n): (88555656917232237327676489978536853396317165487572791298553276065549008783779279197459683778239856445981736945411789064768581524478860917950627181738254 73, 91228320847959651315590708831340204049773457952419290808011790142695905004364637713911427940024014510316500412383083724307748611361219068377046272490145 93)
```

<u>Шифрування та розшифрування (для ключів абонента B):</u>

Обрали ВТ:

```
 Тест RSA (Шифрування)
Відкрите повідомлення M: 123456789012345
```

Виконали зашифрування ($C = M^e (mod\ n_B)$):

```
Шифртекст (криптограма) C: 250545564886131882190459944668869792711182939078754390948591978593229035377207850584662291519470700625920251566973116498915136216605443413081399501342612 9
```

Виконали розшифрування ($M = C^d (mod\ n_B)$):

```
Розшифровано: 123456789012345
```

Результат: вхідне та розшифроване повідомлення співпадають.

Цифровий підпис (для ключів абонента А):

Повідомлення для підпису:

```
 Тест Цифрового Підпису
Повідомлення для підпису M: 987654321
```

Ми сформували підпис ($S = M^{d_A}(mod\ n_A)$):

```
Цифровий підпис S: 463698202344221646186880076411506218647489907188955686750544318907352627907410241074609038724290773396487280756268824683834
558369631413086756551979858048O
```

Та перевірили підпис: ($M' = S^{e}(mod\ n_A)$): Результат True (автентифікація успішна)

```
Результат перевірки підпису:True
```

Протокол конфіденційного розсилання ключів

Ми здійснили передачу секретного ключа k від Абонента А до Абонента В.

Дії Абонента А (відправник):

Ми обрали секретне значення k:

```
Запуск протоколу конфіденційного розсилання ключів
Секретний ключ k: 22868757440419657541544423319639906656665462488739107718006191967754211364993751826134454465905756612976008660547624806590
560725837298262445816896955878
```

Ми сформували цифровий підпис $S = k^{dA}(mod\ n_A)$:

```
Протокол конфіденційного розсилання ключів (Абонент А):
1. Підпис сформовано (S):34348976160188238089556841289319269346062460051572313711313403428139984332719094920467788352783779559261417319666495
592614173196664927293769161102436435 21
```

Ми зашифрували ключ ключем отримувача $S = k^{eB}(mod\ n_B)$:

```
2. k зашифровано (k1): 49728569976887174221273001201874407083652061627927022237195307841622335434631591073933099956667101617829808912804430060
971593197601306777651641872040078 30
```

Ми зашифрували підпис ключем отримувача $S = S^{eB}(mod\ n_B)$:

```
3. S зашифровано (S1): 52852297195848184505293466103455354121991224880645856756382747607911212742858341733757106422711894457928617625863507642
784712191459003737594087211252056 89
```

Ми відправили повідомлення (k,S) абоненту В.

Дії Абонента В (отримувач):

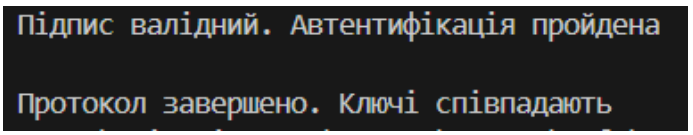
Ми розшифрували значення ключа $k' = k^{dB}(mod\ n_B)$:

```
Протокол конфіденційного розсилання ключів (Абонент В):
  [1] Розшифровано k': 22868757440419657541544423319639906656665462488739107718006191967754211364993751826134454465905756612976008660547624806
590995607258372982624581689695587 8
```

Ми розшифрували значення підпису $S' = S_1^{d_B}(mod\ n_B)$:

```
 [2] Розшифровано S': 34348976160188238089556841289319269346062460051572313711313403428139984332719094920467788352783779544597700365296978592
61417319666492729376916110243643521
```

Ми виконали перевірку підпису $k' \equiv (S')^{e_A}(mod\ n_A)$:



Результат позитивний. Ключ k отримано успішно. Автентичність відправника A підтверджено.

Висновки:

У ході лабораторної роботи ми практично реалізували криптосистему RSA. Ми дослідили роботу ймовірнісних тестів на простоту (Міллера-Рабіна), які дозволяють нам ефективно знаходити великі прості числа. Також розробили програмний комплекс, що виконує генерацію ключів, шифрування, дешифрування та накладання/перевірку цифрового підпису.

Ми продемонстрували роботу протоколу конфіденційного розсилання ключів. Ми експериментально підтвердили, що для коректної передачі підписаного та зашифрованого повідомлення модуль отримувача n_B має бути не меншим за модуль відправника n_A, інакше відновлення підпису неможливе. Реалізований нами протокол забезпечив конфіденційність (завдяки шифруванню на ключі B) та автентичність (завдяки підпису на ключі A).