**УКРАЇНИ МІНІСТЕРСТВО ОСВІТИ І НАУКИ**

**СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ**

**ЦЕНТР ЗАОЧНОЇ, ДИСТАНЦІЙНОЇ ТА ВЕЧІРНЬОЇ ФОРМ НАВЧАННЯ**

**КАФЕДРА КОМП’ЮТЕРНИХ НАУК**

**ПРАКТИЧНА РОБОТА № 3**

з дисципліни "Захист інформації"

Тема: **" Алгоритм RSA "**

Варіант № 13

**Перевірив Коробченко О.В.**

**Студент групи Індн – 31с Кролевецький П.А.**

Суми – 2017

**Мета:**

Виробити вміння та навички шифрування даних за допомогою шифру алгоритму RSA.

**Завдання.**

1. Ознайомитися з алгоритмом шифрування даних RSA.

2. Виконати програмну реалізацію шифрування і дешифрування даних за допомогою алгоритму RSA, згідно свого варіанту.

3. Протестувати роботу програми.

4. Оформити звіт.

|  |  |  |  |
| --- | --- | --- | --- |
| **№**  **в-та** | **A** | **B** | **Алфавіт** |
| **13** | **7000** | **14000** | **Великі букви українського алфавіту і малі букви англійського алфавіту** |

**Теоретична частина**

Алгоритм RSA запропонували в 1978 три автори: Р. Ривест (Rivest), А. Шамір (Shamir) і А. Адлеман (Adleman). Алгоритм отримав свою назву за першими літерами прізвищ його авторів. Алгоритм RSA став першим повноцінним алгоритмом з відкритим ключем, який може працювати як в режимі шифрування даних, так і в режимі електронного цифрового підпису.

Надійність алгоритму ґрунтується на складності факторизації великих чисел і складності обчислення дискретних логарифмів.

У криптосистемі RSA відкритий ключ КА, секретний ключ КВ, повідомлення М і криптограма С належать множині цілих чисел

*,*  (1)

де ***N*** – модуль:

*.* (2)

***Р*** і ***Q*** - випадкові великі прості числа.

Для забезпечення максимальної безпеки вибирають ***Р*** і ***Q*** однакової довжини і зберігають у таємниці.

Множина ***ZN*** з операціями додавання і множення по модулю ***N*** утворює арифметику по модулю ***N***.

Відкритий ключ ***КА*** вибирають випадковим чином так, щоб виконувалися умови:

, (3)

, (4)

де  – функція Ейлера, показує кількість позитивних чисел в інтервалі від 1 до ***N***, які взаємно прості з ***N***:

.

Умова (4) означає, що відкритий ключ ***КА*** і функція Ейлера  повинні бути взаємно простими. Далі, використовуючи розширений алгоритм Евкліда, обчислити секретний ключ ***KB*** з порівняння:

**  (5)

або

.

Це можна здійснити, завдяки тому що одержувач ***В*** знає пару простих чисел (***P, Q***) і може легко знайти . Зауважимо, що ***KB*** і ***N*** повинні бути взаємно простими.

Відкритий ключ ***КА*** використовують для шифрування даних, а секретний ключ ***KB*** - для розшифрування. Перетворення шифрування визначає криптограму ***C*** через пару (відкритий ключ ***КА***, повідомлення ***М***) відповідно до такої формули:

**.** (6)

Звернення функції , тобто визначення значення ***М*** за відомим значенням ***С***, ***КA*** і ***N***, практично нездійсненне при ***N*** > 2512. Однак зворотну задачу, тобто задачу розшифрування криптограми ***C***, можна вирішити, використовуючи пари (секретний ключ ***KB***, криптограма ***C***) за формулою:

.  (7)

Процес розшифрування можна записати так:

. (8)

Підставляючи в (8) значення (6) і (7), одержуємо:

,

або

**.** (9)

Величина  відіграє важливу роль в теоремі Ейлера, яка стверджує: якщо , то

,

або в більш загальній формі

. 10)

Зіставляючи вирази (9) і (10), одержуємо



або, що те ж саме, .

Саме тому для обчислення секретного ключа ***KB*** використовують співвідношення (5). Таким чином, якщо криптограму



піднести до ступеня KB, то в результаті відновлюється вихідний відкритий текст М, так як



Таким чином, одержувач ***В***, що створює криптосистему, захищає два параметри: *секретний ключ* ***KB*** і *пару чисел* (***P, Q***), добуток яких дає значення модуля ***N***. З іншого боку, одержувач ***В*** відкриває значення модуля ***N*** і відкритий ключ ***КА***. Противнику відомі тільки значення ***КА*** і ***N***. Якби він зміг розкласти число ***N*** на множники ***Р*** і ***Q***, то він дізнався б "потайний хід" - трійку чисел {***Р, Q, КA***}, обчисливши значення функції Ейлера



і визначивши значення секретного ключа ***KB***. Однак, як було сказано вище, розкладання дуже великого ***N*** на множники обчислювальної неможливі (за умови, що довжини обраних ***Р*** і ***Q*** становлять не менше 100 десяткових знаків).

Розглянемо алгоритм шифрування і розшифрування в криптосистемі RSA. Припустимо, що користувач ***А*** хоче передати користувачеві ***В*** повідомлення в зашифрованому вигляді, використовуючи криптосистему RSA. У такому випадку користувач ***А*** виступає в ролі відправника, а користувач ***В*** - в ролі одержувача. Як зазначалося вище, криптосистему RSA повинен сформувати одержувач повідомлення, тобто користувач ***В***. Розглянемо послідовність дій користувача ***В*** і користувача ***А:***

**1.** Користувач ***В*** вибирає два довільних великих простих числа ***Р*** і ***Q***;

**2.** Користувач ***В*** обчислює значення модуля ***N***

;

**3.** Користувач ***В*** обчислює функцію Ейлера (8)

;

**4.** Користувач ***В*** вибирає випадковим чином значення відкритого ключа ***КА*** з урахуванням виконання умов:

, ;

**5.** Користувач ***В*** обчислює значення секретного ключа ***КB***, використовуючи розширений алгоритм Евкліда при вирішенні порівняння

;

**6.** Користувач ***В*** пересилає користувачеві ***А*** пару чисел (***N, КA***) по незахищеному каналу.

Якщо користувач ***А*** хоче передати користувачеві ***В*** повідомлення ***М***, він виконує наступні кроки:

* користувач ***А*** розбиває вихідний відкритий текст ***М*** на блоки, кожен з яких може бути представлений у вигляді числа *Мi=*0,1,2,...,*N*-1 ;
* користувач ***А*** шифрує текст, представлений у вигляді послідовності цифр ***М***, за формулою

;

* користувач ***А*** відправляє криптограму

*C*1*, С*2*, С*3,...,*Ci, ...*

користувачу ***В;***

* користувач ***В*** розшифровує прийняту криптограму

*C*1*, С*2*, С*3,...,*Ci, ...,*

використовуючи секретний ключ ***КB***, по формулі

.

У результаті буде отримана послідовність чисел ***Mi***, яка представляє собою вихідне повідомлення ***М***. Щоб алгоритм RSA мав практичну цінність, необхідно мати можливість без істотних витрат генерувати великі прості числа, вміти оперативно обчислювати значення ключів ***КA*** і ***КB***.

**Вихідні дані, які були застосовані для виконання поставленого завдання**

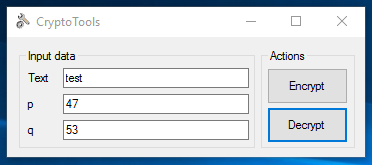
P=47

Q=53

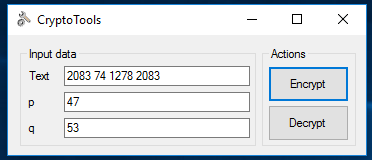
Текст що шифрувався: test

**Скриншоти роботи програми**

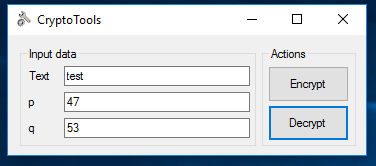
Початкові данні:



Після проведення кодування:



Після декодування:



**Вихідний текст програми** (додається до звіту)

**Результати роботи програми**

P=47

Q=53

Текст що шифрувався: test

Результат: 2083 74 1278 2083

**Висновки**

В ході виконання роботи вироблені навички шифрування даних за допомогою шифру алгоритму RSA.