Лабораторная работа № 4

«Криптографические системы с открытым ключом»

3.1. Цель работы:

Изучить криптографические алгоритмы с открытым ключом. Программно реализовать алгоритм.

3.2. Теоретическая часть

Первые криптографические системы с открытым ключом появились в конце 1970-х годов. От классических алгоритмов они отличаются тем, что для шифрования данных используется один ключ (*открытый*), а для дешифрования – другой (*секретный*). Данные, зашифрованные открытым ключом, можно расшифровать *только* секретным ключом. Следовательно, открытый ключ может распространяться через обычные коммуникационные сети и другие открытые каналы. Таким образом, устраняется главный недостаток стандартных криптографических алгоритмов: необходимость использовать специальные каналы связи для распределения ключей. Разумеется, секретный ключ не может быть вычислен из открытого ключа.

В настоящее время лучшим криптографическим алгоритмом с открытым ключом считается RSA (по имени создателей: Rivest, Shamir, Adelman). Перед изложением метода RSA определим некоторые термины.

Под *простым числом* будем понимать такое число, которое делится только на 1 и на само себя.

Взаимно простыми числами будем называть такие числа, которые не имеют ни одного общего делителя, кроме 1.

Под результатом операции $i \mod j$ будем понимать остаток от целочисленного деления i на j.

Наиболее важной частью алгоритма RSA, как и других алгоритмов с открытым ключом, является процесс создания пары открытый/секретный ключи. В RSA он состоит из следующих шагов.

1. Случайным образом выбираются два секретных простых числа p и q, $p \neq q$.
2. Вычисляется $n=p*q$.
3. Вычисляется $\varphi=(p-1)*(q-1)$.
4. Выбираются открытый (K_o) и секретный (K_c) ключи, которые являются взаимно простыми с φ и удовлетворяют условию $(K_o*K_c) \bmod \varphi = 1$.

Чтобы зашифровать данные открытым ключом K_o, необходимо:

1) разбить исходный текст на блоки, каждый из которых может быть представлен в виде числа $M(i)=2,n-1$;

2) зашифровать последовательность чисел M(i) по формуле

$$C(i) = (M(i)^{K_o}) \bmod n,$$

где последовательность чисел C(i) представляет шифротекст.

Чтобы расшифровать эти данные секретным ключом K_c, необходимо выполнить следующие вычисления:

$$M(i) = (C(i)^{K_c}) \bmod n.$$

В результате будет получено множество чисел M(i), которые представляют собой исходный текст.

Приведем простой пример использования метода RSA для шифрования сообщения «CAB». Для простоты будем использовать малые числа (на практике используются намного большие числа).

1. Выберем p=3, q=11.

2. Вычислим n=3*11=33.
3. Вычислим $\varphi=(p-1)*(q-1)=20$.
4. Выберем секретный ключ K_c, который является взаимно простым с φ, например $K_c=3$.
5. На основе K_c и φ вычислим открытый ключ K_o. Для этого можно использовать расширение алгоритма Евклида:

BEGIN

    $g0=\varphi$; $g1=K_c$;
    u0=1; u1=0;
    v0=0; v1=1;
    i=1;
        while $g_i \neq 0$ do
            begin
                $g_i=u_i\ \varphi + v_i\ K_c$;
                $y=g_{i-1}$ div g_i;
                $g_{i+1}=g_{i-1} - yg_i$;
                $u_{i+1}=u_{i-1} - yu_i$;
                $v_{i+1}=v_{i-1} - yv_i$;
                i=i+1;
            end;
    $K_o=v_{i-1}$;
    if $K_o<0$ then $K_o=K_c\ \ \varphi$;

END.

В соответствии с алгоритмом получаем $K_o=7$.

6. Представим шифруемое сообщение как последовательность целых чисел в диапазоне 2…28. Пусть букве A соответствует число 3, букве B – число 4, а букве C – число 5. Тогда сообщение «CAB» можно представить виде последовательности чисел {5,3,4}. Зашифруем сообщение, используя открытый ключ $K_o=7$:

$C1 = (5^7) \bmod 33 = 78125 \bmod 33 = 14$,
$C2 = (3^7) \bmod 33 = 2187 \bmod 33 = 9$,
$C3 = (4^7) \bmod 33 = 16384 \bmod 33 = 16$.

7. Для расшифровки полученного сообщения {14,9,16} с помощью секретного ключа $K_c=3$ необходимо:

$M1 = (14^3) \bmod 33 = 2744 \bmod 33 = 5$,
$M2 = (9^3) \bmod 33 = 729 \bmod 33 = 3$,
$M3 = (16^3) \bmod 33 = 4096 \bmod 33 = 4$.

Таким образом в результате дешифрования сообщения получено исходное сообщение {5,3,4} («CAB»).

Криптостойкость алгоритма RSA основывается на предположении, что исключительно трудно определить секретный ключ по открытому, поскольку для этого необходимо решить задачу о существовании делителей целого числа. Данная задача является NP-полной, то есть не имеет эффективного (полиномиального) решения. Вопрос существования эффективных алгоритмов решения NP-полных задач является до настоящего времени открытым. Традиционные же методы для чисел, состоящих из 200 цифр (именно такие числа рекомендуется использовать), требуют выполнения огромного числа операций (около 10^{23}).

3.3. Порядок выполнения работы

1. Реализовать алгоритм генерирования простых чисел.
2. Написать программу проверки взаимной простоты чисел на основе бинарного алгоритма и алгоритма Евклида.
3. Реализовать алгоритм RSA (шифратор и дешифратор).
4. Оформить лабораторную работу на листах формата А4 в соответствии с правилами оформления студенческих лабораторных работ. Отчет по лабораторной работе должен содержать:

– титульный лист (оформленный по правилам) с указанием названия работы и следующие разделы:
– цель работы;
– краткие теоретические сведения;
– задание на выполнение работы;
– алгоритм решения поставленной задачи (блок-схема и словесное описание по блокам);
– текст программы;
– описание входных / выходных данных программы;
– контрольный пример;
– выводы;
– список использованной литературы.

Необходимо разработать четкий и удобный интерфейс пользователя для задания исходных данных и вывода результатов работы программы.