# Лабораторная работа №4

**АЛГОРИТМ RSA**

# Ассиметричные алгоритмы шифрования данных

***Цель работы***:освоить методику работы ассиметричных алгоритмов шифрования, где суще­ствует два ключа – один для шифрования, другой для дешифрования.

**Теоретические сведения**

Алгоритм RSA разработан в 1977 г. Роном Ривестом, Ади Шамиром и Ле­ном Адлеманом и опубликован в 1978 г. С тех пор алгоритм Rivest-Shamir-Adleman (RSA) широко применяется практически во всех приложениях, исполь­зующих криптографию с открытым ключом.

Алгоритм RSA:

*1. Вычисление ключей*

Важным моментом в этом криптоалгоритме является создание пары ключей: открытого и закрытого. Для алгоритма RSA этап создания ключей состоит из сле­дующих операций:

1.1. Выбираются два простых различных числа *p* и *q*. Вычисляется их произ­ведение *n* = *p · q*, называемое модулем. 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 единицы.

1.2. Вычисляется функция Эйлера *Ф*(*n*) = (*p* – 1) *·* (*q* – 1).

1.3. Выбирается произвольное число *e* (*e* < *n*), такое, что 1 < *e* < *Ф*(*n*) и не имеет общих делителей, кроме 1 (взаимно простое) с числом (*p* – 1) *·* (*q* – 1).

1.4. Вычисляется *d* методом Евклида таким образом, что (*e · d* – 1) де­лится на (*p* – 1) *·* (*q* – 1).

1.5. Два числа (*е*, *n*) публикуются как открытый ключ.

1.6. Число *d* хранится в секрете – закрытый ключ есть пара (*d*, *n*), кото­рый позволит читать все послания, зашифрованные с помощью пары чисел (*е*, *n*).

*2. Шифрование*

Шифрование с помощью пары чисел производится следующим образом:

2.1. Отправитель разбивает своё сообщение *M* на блоки *mi*. Значение *mi < n*, поэтому длина блока *mi* в битах не больше *k* = [log2(*n*)] бит, где квадратные скобки обозначают, взятие целой части от дробного числа.

Например, если *n* = 21, то максимальная длина блока *k* = [log2(21)] = [4.39…]= = 4 бита.

2.2. Подобный блок может быть интерпретирован как число из диапазона (0; 2*k*– 1). Для каждого такого числа *mi* вычисляется выражение (*ci* – зашифро­ванное сообщение): *ci* = ((*mi*)*e*) mod *n*.

Необходимо добавлять нулевые биты слева в двоичное представление блока  до размера *k* = [log2(*n*)] бит.

*3. Дешифрование*

Чтобы получить открытый текст, необходимо каждый блок дешифровать отдельно: *mi* = ((*ci*)*d*) mod *n*.

*Пример:*

Выбрать два простых числа: *р* = 7, *q* = 17.  
Вычислить *n* = *p* *· q* = 7 *·* 17 = 119.  
Вычислить *Ф*(*n*) = (*p* – 1) *·* (*q* – 1) = 96.

Выбрать *е* так, чтобы *е* было взаимнопростым с *Ф*(*n*) = 96 и меньше, чем *Ф*(*n*): *е* = 5.

Определить *d* так, чтобы *d · e* ≡ 1 mod 96 и *d* < 96, *d* = 77, так как

77 *·* 5 = 385 = 4 *·* 96 + 1.

Результирующие ключи открытый {5, 119} и закрытый ключ {77, 119}.

Например, требуется зашифровать сообщение *М* = 19: 195 = 66 (mod 119),

*С* = = 66. Для дешифрования вычисляется 6677 (mod 119) = 19.

**Варианты заданий**

**(Красное не делать)**

1. Разработать консольное приложение для шифрования/дешифрования произвольных фай­лов с помощью алгоритма RSA.

2. Разработать визуальное приложение для шифрования/дешифрования изо­бражений.

3. Разработать визуальное приложение для шифрования/дешифрования про­извольных файлов.

4. Разработать клиент-серверное приложение для защищённой передачи фай­лов по сети.

5. Разработать клиент-серверное приложение для защищённого обмена сооб­щениями по сети.

6. Разработать визуальное приложение для шифрования/дешифрования чисел.

7. Разработать консольное приложение для генерации ключей.

8. Реализовать программу для шифрования / дешифрования текстов, работающую по алгоритму RSA. Программа должна уметь работать с текстом произвольной длины.

**Контрольные вопросы:**

1. Дайте определение алгоритма с открытым ключом.

2. Сколько этапов содержит алгоритм RSA?

3. В чем заключается вычисление ключей алгоритма RSA?

4. Как происходит шифрование в алгоритме RSA?

5. Как происходит дешифрование в алгоритме RSA?