ЛАБОРАТОРНАЯ РАБОТА 2
РЕАЛИЗАЦИЯ ЭЛЕМЕНТОВ КРИПТОСИСТЕМЫ RSA

Цель работы: формирование умений шифрования с использованием метода асимметрического шифрования RSA.

Теоретические сведения

RSA относится к так называемым *асимметричным алгоритмам*, у которых ключ шифрования не совпадает с ключом рас шифровки. Один из ключей доступен всем и называется открытым ключом, другой хранится только у хозяина и никому неизвестен. С помощью одного ключа можно производить операции только в одну сторону. Если сообщение зашифровано с помощью одного ключа, то расшифровать его можно только с помощью другого. Имея один из ключей практически невозможно найти другой ключ, если разрядность ключа высока.

Описание RSA

Алгоритм RSA состоит из следующих пунктов:

1. Выбрать простые числа p и q.
2. Вычислить n = pq.
3. Вычислить m = (p – 1) • (q – 1).
4. Выбрать число d взаимно простое с m.
5. Выбрать число e так, чтобы ed = 1 (mod m).

Числа e и d являются ключами RSA. Шифруемые данные необходимо разбить на блоки – числа от 0 до n – 1. Шифрование и расшифровка данных производятся следующим образом:

- шифрование: $b = a^e \pmod n$;
- расшифрование: $a = b^d \pmod n$.

Следует также отметить, что ключи e и d равноправны, то есть сообщение можно шифровать как ключом e, так и ключом d, при этом расшифровка должна быть произведена с помощью другого ключа.

Нахождение простых чисел

В первом пункте алгоритма RSA сказано, что необходимо выбрать два простых числа p и q. Простой способ – деление предполагаемого простого числа на все числа меньшие его (не работоспособен уже с 32-битными числами, поскольку требуется большое количество времени на выполнение).

В данном случае, для выработки простых чисел используют вероятностные методы, но они не дают полной гарантии, что найденное число простое. Однако при достаточно небольшом количестве операций можно получить высокую вероятность, что найденное число является простым.

Алгоритм поиска простых чисел

1. N – нечетное число. Найти s и t, удовлетворяющие уравнению: $N - 1 = 2^s \cdot t$.

2. Случайным образом выбрать число a, $1 < a < N$.

3. Если N делится на a, перейти к пункту 6.

4. Если условие $at = 1 \pmod N$ выполняется, перейти к пункту 2.

5. Если найдется такое k, $0 \le k < s$, что $a^{2kt} = -1 \pmod N$, перейти к пункту 2.

6. Число N – составное: выбрать другое нечетное число N, перейти к пункту 1.

Если для какого-либо числа N проверено m чисел a, то математически доказанная вероятность того, что число является составным будет равняться 4 – m (но вероятность намного меньше этого значения). Исходя из этого, для числа N, состоящего из p бит, проверить p различных значений a. Если во время этого не обнаружится, что N – число составное, то вероятно, что оно является простым.

Нахождение взаимно простых чисел

На шаге 4 алгоритма RSA необходимо найти число d взаимно простое с m, то есть не имеющее общих делителей с ним, кроме единицы. Число d должно быть меньше m, таким образом, разрядность числа d равна сумме бит в числах p и q. Для нахождения взаимно простых чисел используется алгоритм Евклида, который находит наибольший общий делитель двух чисел. Если

найденный делитель больше единицы, то необходимо выбрать другое число d и повторить проверку.

Алгоритм Евклида

1. Исходные числа a и b.
2. Вычислить r – остаток от деления a на b: a = bq + r.
3. Если r = 0, то b – искомое число (наибольший общий делитель), конец.
4. Если пункт 3 не выполняется, заменить пару чисел <a, b> парой <b, r>, перейти к пункту 2.

При вычислении наибольшего общего делителя с помощью алгоритма Евклида будет выполнено не более 5p операций деления с остатком, где p есть количество цифр в десятичной записи меньшего из чисел a и b.

Решение уравнения ax + by = 1

В 5-м пункте алгоритма RSA предполагается нахождение такого числа e, чтобы ed = 1 (mod m). Для этого нужно использовать модифицированный алгоритм Евклида, который работает только если числа d и m взаимно просты. Вычисление числа e сводится к решению уравнения mx + de = 1 в натуральных числах. Число x не существенно.

Алгоритм решения уравнения ax + by = 1

1. Необходимо определить матрицу $E = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$.
2. Вычислить r – остаток от деления a на b: a = bq + r.
3. Если r = 0, то второй столбец матрицы дает решение: $\begin{bmatrix} x \\ y \end{bmatrix}$, конец.
4. Если пункт 3 не выполняется, то вычислить $E = E \begin{bmatrix} 0 & 1 \\ 1 & -q \end{bmatrix}$.
5. Заменить пару чисел <a, b> парой <b, r>, перейти к пункту 2.

В данном алгоритме все вычисления можно производить по модулю большего из чисел a и b. Отрицательное число – q заменяется положительным, полученным путем вычитания числа q из числа, взятого в качестве модуля. Например, если из чисел a и b большим является число b, то все вычисления можно производить по модулю числа b, при этом – q

будет представлено как b – q. Скорость работы алгоритма и количество производимых им операций примерно равняется соответствующим параметрам алгоритма Евклида, описанного выше.

Большие числа и работа с ними

На данный момент времени рекомендуется в качестве чисел e и d брать числа, длиной не менее 768 бит. Ключ в 1024 бит является достаточно надежным для обычных целей шифрования. Для повышенной безопасности рекомендуется брать ключи размером 2048 бит, то есть числа p и q должны иметь разрядность вдвое ниже чисел e, d, m и n (p и q рекомендуется брать примерно одного порядка, но не слишком близко друг к другу).

Содержание заданий

Разработайте программу, имитирующую реализацию элементов метода криптографической защиты информации RSA. Программа должна выполнять генерацию ключей, шифрование и расшифрование сообщения. В качестве сообщения используйте свою фамилию. Использовать n длиной в 4 разряда и более.

Контрольные вопросы

1. Что такое ассиметричное шифрование?
2. Отличие ассиметричного шифрования от блочного?
3. Перечислите преимущества и недостатки алгоритма RSA?

Отчетность по лабораторной работе

Включает код программы с подробными его комментариями и результатами выполнения программы.