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 $n = p \cdot q$, 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 $\Phi(n) = (p - 1) \cdot (q - 1)$.

1.3. Выбирается произвольное число e ($e < n$), такое, что $1 < e < \Phi(n)$ и не имеет общих делителей, кроме 1 (взаимно простое) с числом $(p - 1) \cdot (q - 1)$.

1.4. Вычисляется d методом Евклида таким образом, что $(e \cdot d - 1)$ делится на $(p - 1) \cdot (q - 1)$.

1.5. Два числа (e, 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 (e, n).

2. Шифрование

Шифрование с помощью пары чисел производится следующим образом:

2.1. Отправитель разбивает своё сообщение M на блоки m_i. Значение $m_i < n$, поэтому длина блока m_i в битах не больше $k = [\log_2(n)]$ бит, где квадратные скобки обозначают, взятие целой части от дробного числа.

Например, если $n = 21$, то максимальная длина блока $k = [\log_2(21)] = [4.39\ldots]= = 4$ бита.

2.2. Подобный блок может быть интерпретирован как число из диапазона $(0; 2^k - 1)$. Для каждого такого числа m_i вычисляется выражение (c_i – зашифрованное сообщение): $c_i = ((m_i)^e) \bmod n$.

Необходимо добавлять нулевые биты слева в двоичное представление блока c_i до размера $k = [\log_2(n)]$ бит.

3. Дешифрование

Чтобы получить открытый текст, необходимо каждый блок дешифровать отдельно: $m_i = ((c_i)^d) \bmod n$.

Пример:

Выбрать два простых числа: $p = 7$, $q = 17$.

Вычислить $n = p \cdot q = 7 \cdot 17 = 119$.

Вычислить $\Phi(n) = (p - 1) \cdot (q - 1) = 96$.

Выбрать e так, чтобы e было взаимнопростым с $\Phi(n) = 96$ и меньше, чем $\Phi(n)$: $e = 5$.

Определить d так, чтобы $d \cdot e \equiv 1 \bmod 96$ и $d < 96$, $d = 77$, так как $77 \cdot 5 = 385 = 4 \cdot 96 + 1$.

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

Например, требуется зашифровать сообщение $M = 19$: $19^5 = 66 \pmod{119}$, $C = = 66$. Для дешифрования вычисляется $66^{77} \pmod{119} = 19$.

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?