Лабораторная работа № 6

Асимметричный алгоритм шифрования RSA

Цель работы

Изучить принцип работы асимметричных алгоритмов шифрования на примере алгоритма RSA. Освоить методику создания комбинированных алгоритмов шифрования, которые совмещают достоинства методов симметричной и асимметричной криптографии.

Алгоритм шифрования RSA

Алгоритм RSA был разработан в 1977 году Роном Ривестом, Ади Шамиром и Леном Адлеманом и опубликован в 1978 году. С тех пор алгоритм *Rivest-Shamir-Adleman* (RSA) широко применяется практически во всех приложениях, использующих криптографию с открытым ключом [1, 2, 3].

Алгоритм RSA состоит из трёх этапов:

I. ***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. Выбираются два простых различных числа p и q. Вычисляется их произведение $n = p \cdot q$, называемое ***модулем***.
   2. Вычисляется функция Эйлера $\varphi(n) = (p-1) \cdot (q-1)$.
   3. Выбирается произвольное число e ($e < n$) такое, что $1 < e < \varphi(n)$ и e не имеет с числом $\varphi(n)$ других общих делителей, кроме 1 (т.е. оно является взаимно простым с ним).
   4. Вычисляется d (алгоритмом Евклида) таким образом, что $(e \cdot d - 1)$ делилось на $\varphi(n)$.
   5. Два числа $(e,\ n)$ публикуются как ***открытый ключ***.
   6. Число d хранится в секрете. Пара $(d,\ n)$ есть ***закрытый ключ***, который позволит читать все послания, зашифрованные с помощью пары чисел $(e,\ n)$.

II. ***Шифрование*** с помощью этих ключей производится следующим образом:
   1. Отправитель разбивает своё сообщение M на блоки m_i. Значение $m_i < n$, поэтому длина блока открытого текста 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 открытого текста $k = [\log_2 21] = [4.39\ldots] = 4$ бита.
   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):

$$c_i = (m_i)^e \bmod n.$$

      В качестве размера блока зашифрованного текста следует брать $k_e = \lceil \log_2 n \rceil$ бит, где операция $\lceil\ \rceil$ – это округление вверх до ближайшего целого.
      Необходимо добавлять нулевые биты слева в двоичное представление блока c_i до размера k_e бит.

III. ***Дешифрование*** производится следующим образом:
   1. Чтобы получить открытый текст, надо каждый блок зашифрованного текста длиной k_e бит 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 $\varphi(n) = (p-1) \cdot (q-1) = 96$.
Выбрать e так, чтобы e было взаимно простым с $\varphi(n) = 96$ и меньше, чем $\varphi(n)$: $e = 5$.
Определить d так, чтобы $d \cdot e \equiv 1 \bmod 96$ и $d < 96$: $d = 77$, т.к. $77 \cdot 5 = 385 = 4 \cdot 96 + 1$.

Результирующие ключи: открытый ключ $(5,\ 119)$ и закрытый ключ $(77,\ 119)$.
Пусть, например, требуется зашифровать сообщение $M = 19$:

$$C = 19^5 = 66 \pmod{119}.$$

Для дешифрования вычисляется $66^{77} \pmod{119} = 19$.

Комбинирование симметричных и асимметричных алгоритмов

Симметричные алгоритмы и, в частности, DES – быстрые, поэтому ими удобно шифровать большие объёмы информации. Однако для передачи ключа симметричного алгоритма требуется надёжный канал передачи, который очень часто отсутствует. Таким образом, преимущества таких алгоритмов сводятся на нет. С другой стороны, асимметричные алгоритмы не требуют секретного канала для передачи ключа, но на практике криптосистемы с открытым ключом используются для шифрования не сообщений, а ключей. На это есть две основные причины:

1. Алгоритмы шифрования с открытым ключом в среднем работают в тысячи раз медленнее, чем симметричные алгоритмы, а также они требовательны к памяти и вычислительной мощности компьютера, поэтому большие тексты кодировать этими алгоритмами нецелесообразно.
2. Алгоритмы шифрования с открытым ключом уязвимы по отношению к криптоаналитическим атакам со знанием открытого текста. Пусть $C = E(P)$, где C обозначает шифртекст, P – открытый текст, E – функцию шифрования. Тогда, если P принимает значения из некоторого конечного множества, состоящего из n открытых текстов, криптоаналитику достаточно зашифровать все эти тексты, используя известный ему открытый ключ, и сравнить результаты с C. Ключ таким способом ему вскрыть не удастся, однако открытый текст будет успешно определён.

Возможно следующее решение: сообщение шифруется симметричным алгоритмом, что позволяет выиграть в скорости, т.к. сообщение может быть сколь угодно большим, а ключ симметричного алгоритма (обычно маленький, для DES – 56 бит) шифруется асимметричным алгоритмом [1].

Задание

Результатом данной лабораторной работы должны стать приложения, совмещающие в себе достоинства симметричных и асимметричных методов шифрования.

I. Реализовать приложение для шифрования, позволяющее выполнять следующие действия:
   1. Вычислять открытый и закрытый ключи для алгоритма RSA:
      1) числа p и q генерируются программой или задаются из файла;
      2) числа p и q должны быть больше, чем 2^{128};
      3) сгенерированные ключи сохраняются в файлы: открытый ключ $(e,\ n)$ – в один файл, закрытый $(d,\ n)$ – в другой.
   2. Шифровать указанным в варианте симметричным алгоритмом открытый текст, а асимметричным – ключ симметричного алгоритма:
      1) шифруемый текст T должен храниться в одном файле, открытый ключ $(e,\ n)$ для алгоритма RSA – в другом;
      2) ключ K для симметричного алгоритма должен генерироваться случайным образом;
      3) зашифрованный текст должен сохраняться в одном файле, а зашифрованный асимметричным алгоритмом ключ K симметричного алгоритма – в другом;
      4) в процессе шифрования предусмотреть возможность просмотра и изменения шифруемого текста в шестнадцатеричном и символьном виде;
      5) программа должна уметь работать с текстом произвольной длины.

II. Реализовать приложение для дешифрования.
   1. Зашифрованный текст должен храниться в одном файле, зашифрованный ключ симметричного алгоритма – в другом, а секретный ключ для алгоритма RSA – в третьем.

2. Приложение расшифровывает зашифрованный ключ K с помощью алгоритма RSA, а затем с помощью симметричного алгоритма с ключом K расшифровывает зашифрованный текст.
3. Расшифрованный текст должен сохраняться в файл.
4. В процессе дешифрования предусмотреть возможность просмотра и изменения зашифрованного текста в шестнадцатеричном и символьном виде.
5. Программа должна уметь работать с текстом произвольной длины.

III. С помощью реализованных приложений выполнить следующие задания.
1. Протестировать правильность работы разработанных приложений.
2. Сделать выводы о проделанной работе.

Дополнительные критерии оценивания качества работы

1. Наглядность приложений:
   1 – приложения позволяют просматривать и изменять ключи, шифруемый и зашифрованный тексты во всех предусмотренных заданием представлениях;
   0 – приложения позволяют просматривать ключи, шифруемый и зашифрованный тексты только в каком-то одном представлении;
   л.р. не принимается – иначе.

Варианты

Бригады с нечётным номером в качестве симметричного алгоритма должны использовать алгоритм DES, а бригады с чётным – ГОСТ.

Вопросы для защиты

I. **Первая часть защиты** (обязательная):
1. В чём заключается алгоритм RSA?
2. Для чего и почему используют комбинированные криптоалгоритмы?
3. В чём заключаются достоинства и недостатки асимметричных алгоритмов?
4. В чём заключаются достоинства и недостатки симметричных алгоритмов?

II. **Вторая часть защиты**: Найти алгоритмом Евклида элемент d такой, что $e \cdot d \equiv 1 \pmod{n}$, если:
1. $e = 15, n = 82$;
2. $e = 58, n = 115$;
3. $e = 29, n = 86$;
4. $e = 24, n = 95$;
5. $e = 49, n = 122$;
6. $e = 18, n = 107$.

Список литературы

1. Мао, В. Современная криптография: теория и практика : Пер. с англ. / В. Мао. – М. : Издательский дом "Вильямс", 2005. – 768 с.
2. Шнайер, Б. Прикладная криптография. Протоколы, алгоритмы, исходные тексты на языке Си / Б. Шнайер. – М. : Триумф, 2002. – 816 с.
3. PKCS #1 v2.1: RSA Cryptography Standard. – Bedford : RSA Laboratories, 2002. – 61 p.