МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования
«ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ»

В.Г. Спицын

Методические указания к выполнению лабораторных работ по дисциплине “Защита информации”

Томск 2012

Методические указания к выполнению лабораторных работ
"Разработка программ шифрования на основе методов Полибия, замены, умножения матриц, криптоалгоритма RSA и комплекса криптоалгоритмов PGP"

Целью лабораторных работ является создание студентами программных реализаций шифрования на основе методов Полибия, замены, умножения матриц, криптоалгоритма RSA и комплекса криптоалгоритмов PGP.
Разработанные программы шифрования демонстрируются преподавателю на примере тестовых задач.
Отчет по лабораторной работе должен содержать:
1. Цель работы.
2. Постановку задачи.
3. Метод решения задачи.
4. Структурную схему алгоритма.
5. Листинг программы.
6. Результаты работы программ.
7. Выводы.

1. Шифрование и дешифрирование информации

Криптография дает возможность преобразовать информацию таким образом, что ее прочтение (восстановление) возможно только при знании ключа.

В качестве информации, подлежащей шифрованию и дешифрированию, будут рассматриваться *тексты*, построенные на некотором *алфавите*. Под этими терминами понимается следующее.

Алфавит – конечное множество используемых для кодирования информации знаков.

Текст – упорядоченный набор из элементов алфавита.

Шифрование – преобразовательный процесс: *исходный текст*, который носит также название *открытого текста*, заменяется *шифрованным текстом* (рис. 1).

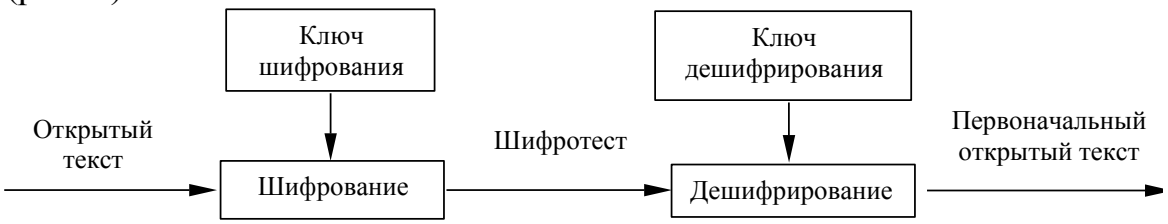

Рис. 1.

Дешифрирование – обратный шифрованию процесс. На основе ключа шифрованный текст преобразуется в исходный (рис. 1).

Ключ – информация, необходимая для беспрепятственного шифрования и дешифрирования текстов.

Криптографическая система представляет собой семейство **T** преобразований открытого текста. Пространство ключей *K* – это набор возможных значений ключа. Обычно ключ представляет собой последовательный ряд букв алфавита.

Криптосистемы разделяются на *симметричные* и *с открытым ключом*.

В *симметричных криптосистемах* и для шифрования, и для дешифрирования используется *один и тот же ключ*.

В *системах с открытым ключом* используются два ключа – *открытый* и *закрытый*, которые математически связаны друг с другом. Информация шифруется с помощью открытого ключа, который доступен всем желающим, а расшифровывается с помощью закрытого ключа, известного только получателю сообщения.

Термины *распределение ключей* и *управление ключами* относятся к процессам системы обработки информации, содержанием которых является составление и распределение ключей между пользователями.

Электронной (цифровой) подписью называется присоединяемое к тексту его криптографическое преобразование, которое позволяет при получении текста другим пользователем проверить авторство и подлинность сообщения. *Криптостойкостью* называется характеристика шифра, определяющая его стойкость к дешифрированию без знания ключа (т.е. криптоанализу). Имеется несколько показателей криптостойкости, среди которых

- количество всех возможных ключей;
- среднее время, необходимое для криптоанализа.

Эффективность шифрования с целью защиты информации зависит от сохранения тайны ключа и криптостойкости шифра.

2. Реализация методов защиты информации

Проблема реализации методов защиты информации имеет два аспекта:

- разработку средств, реализующих криптографические алгоритмы;
- методику использования этих средств.

Каждый из рассмотренных криптографических методов может быть реализован либо программным, либо аппаратным способом.

Возможность программной реализации обусловливается тем, что все методы криптографического преобразования формальны и могут быть представлены в виде конечной алгоритмической процедуры.

При аппаратной реализации все процедуры шифрования и дешифрирования выполняются специальными электронными схемами. Основным достоинством программных методов реализации защиты является их гибкость, т.е. возможность быстрого изменения алгоритмов шифрования.

Основным же недостатком программной реализации является существенно меньшее быстродействие по сравнению с аппаратными средствами (примерно в 10 раз).

В последнее время стали появляться комбинированные средства шифрования, так называемые программно-аппаратные средства. В этом случае в компьютере используется своеобразный “криптографический сопроцессор” – вычислительное устройство, ориентированное на выполнение криптографических операций (сложение по модулю, сдвиг и т.д.). Меняя программное обеспечение для такого устройства, можно выбирать тот или иной метод шифрования. Такой метод объединяет в себе достоинства программных и аппаратных методов.

Таким образом, выбор типа реализации криптозащиты информационной системы в существенной мере зависит от ее особенностей и должен опираться на всесторонний анализ требований, предъявляемых к системе защиты информации.

Ниже рассматривается программные реализации криптографических методов:

– метод Полибия;
– метод замены;
– умножение матриц;
– асимметричный алгоритм шифрования данных RSA.

3. Метод Полибия

Методические указания

Шифровальная таблица Полибия представляла собой квадрат с пятью столбцами и пятью строками, которые нумеруются цифрами от 1 до 5. В каждую клетку такого квадрата записывалась одна буква. В результате каждой букве соответствовала пара чисел и шифрование сводится к замене буквы парой чисел.

Идею квадрата Полибия проиллюстрируем таблицей с русскими буквами (Табл. 1). Число букв в русском алфавите отличается от числа букв в греческом алфавите, поэтому и размер таблицы выбран иным (не квадрат 5х5, а прямоугольник 8х4).

Таблица 1

№	1	2	3	4	5	6	7	8
1	А	Б	В	Г	Д	Е	Ж	З
2	И	Й	К	Л	М	Н	О	П
3	Р	С	Т	У	Ф	Х	Ц	Ч
4	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я

Зашифруем фразу: КРИПТОГРАФИЯ:

23 31 21 28 33 27 14 31 11 35 11 35 21 48.

В данном примере видно, что в шифрограмме первым указывается номер строки, а вторым – номер столбца.

Задание

1. Заполнить прямоугольник Полибия, в котором нужно отобразить все буквы русского алфавита от *а* до *я* и от *А* до *Я* плюс символы: пробел, точка, двоеточие, восклицательный знак, вопросительный знак и запятая (всего 72 символа).
2. Методом Полибия зашифровать любую фразу, введенную с клавиатуры.
3. Расшифровать полученную в пункте 2 зашифрованную строку.

4 Разработка программы шифрования на основе метода замены

Методические указания

Шифрование методом замены (подстановки) основано на алгебраической операции, называемой подстановкой.

Подстановкой называется взаимно однозначное отображение некоторого конечного множества M на себя. Число N элементов этого множества называется степенью подстановки. Природа множества M роли не играет, поэтому можно считать, что M={1,2,...,N}.

Если при данной подстановке S число j переходит в I_j, то подстановка обозначается символом

$$S=\begin{bmatrix} 1 & 2 & . & . & n \\ I_1 & I_2 & . & . & I_n \end{bmatrix}.$$

В этой записи числа $1,2,..,n$ можно произвольным образом переставлять, соответственно переставляя числа $I_1, I_2,.., I_n$.

Идею метода замены проиллюстрируем таблицей 2 с русскими буквами. [1].

Таблица 2

А	Б	В	Г	Д	Е	Ё	Ж	З	И	Й
01	02	03	04	05	06	07	08	09	10	11

К	Л	М	Н	О	П	Р	С	Т	У	Ф
12	13	14	15	16	17	18	19	20	21	22

Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я
23	24	25	26	27	28	29	30	31	32	33

Основным недостатком рассмотренного метода является то, что статистические свойства открытого текста (частоты повторения букв) сохраняются в шифротексте.

Общая формула моноалфавитной замены выглядит следующим образом:

$$Y_i = (k_1 \cdot X_i + k_2) \bmod n,$$

где Y_i – i-й символ алфавита; k_1 и k_2 – константы, X_i – i-й символ открытого текста (номер буквы в алфавите); n-длина используемого алфавита.

Шифр, задаваемый формулой

$$Y_i = (X_i + k_i) \bmod n,$$

где k_i – i-я буква ключа, в качестве которого используются слово или фраза, называется шифром Виженера.

Пример

Открытый текст: "*ЗАМЕНА* ".

Ключ: "*КЛЮЧ* ".

$$Y_1 = 9 + 12 (\bmod 33) = 21 \to У,$$
$$Y_2 = 1 + 13 (\bmod 33) = 14 \to М,$$
$$Y_3 = 14 + 32 (\bmod 33) = 13 \to Л,$$
$$Y_4 = 6 + 25 (\bmod 33) = 31 \to Э,$$
$$Y_5 = 15 + 12 (\bmod 33) = 27 \to Щ,$$
$$Y_6 = 1 + 13 (\bmod 33) = 14 \to М.$$

Зашифрованный текст имеет вид "*УМЛЭЩМ* ".

Задание

1. Заполнить таблицу 2 в массив, в котором должны храниться все буквы русского алфавита от *а* до *я* и от *А* до *Я* плюс символы: пробел, точка, двоеточие, восклицательный знак, вопросительный знак и запятая (всего 72 символа).
2. Зашифровать любую фразу с любым ключом (фраза и ключ вводятся с клавиатуры) методом замены.
4. Расшифровать полученный шифротекст, используя тот же ключ.

5. Разработка программы шифрования на основе метода умножения матриц"

Метод умножения матриц использует преобразования следующего вида:

$$Y_i = C \cdot X,$$

где

$$Y = \|y_1, y_2, ..., y_n\|,$$

$$C = \|C_{ij}\|,$$

$$X = \|x_1, x_2, ..., x_n\|.$$

Идею метода умножения матриц также можно проиллюстрировать таблицей с русскими буквами (табл. 2).

Пример

Открытый текст: "ПРИКАЗ" (согласно таблице 2 представим в виде "17 18 10 12 01 09").

Полагаем, что матрица C , представляющая собой ключ шифрования, имеет вид

$$C = \begin{bmatrix} 1 & 3 & 2 \\ 2 & 1 & 5 \\ 3 & 2 & 1 \end{bmatrix}.$$

Процедура получения зашифрованного текста представлена ниже.

$$Y_1 = \begin{bmatrix} 1 & 3 & 2 \\ 2 & 1 & 5 \\ 3 & 2 & 1 \end{bmatrix} \cdot \begin{bmatrix} 17 \\ 18 \\ 10 \end{bmatrix} = \begin{bmatrix} 91 \\ 102 \\ 97 \end{bmatrix},$$

$$Y_2 = \begin{bmatrix} 1 & 3 & 2 \\ 2 & 1 & 5 \\ 3 & 2 & 1 \end{bmatrix} \cdot \begin{bmatrix} 12 \\ 01 \\ 09 \end{bmatrix} = \begin{bmatrix} 33 \\ 70 \\ 47 \end{bmatrix}.$$

В результате шифрования открытого текста имеем: “91 102 97 33 70 47”.

Задание

1. Заполнить таблицу 2 в массив, в котором должны храниться все буквы русского алфавита от *а* до *я* и от *А* до *Я* плюс символы пробел, точка, двоеточие, восклицательный знак, вопросительный знак и запятая (всего 72 символа).
2. Зашифровать любое сообщение, введенное с клавиатуры, методом произведения матриц.
3. Определить какой должна быть матрица, чтобы зашифрованную фразу можно было расшифровать.
4. Расшифровать сообщение.

6. Разработка программной реализации асимметричного алгоритма шифрования данных RSA

Как бы ни были сложны и надежны криптографические системы - их слабое место при практической реализации - проблема *распределения ключей*. Для того, чтобы был возможен обмен конфиденциальной

информацией между двумя субъектами информационной системы (ИС), ключ должен быть сгенерирован одним из них, а затем каким-то образом опять же в конфиденциальном порядке передан другому. Т.е. в общем случае для передачи ключа требуется использование какой-то криптосистемы.

Для решения этой проблемы на основе результатов, полученных классической и современной алгеброй, были предложены системы с открытым ключом. Суть их состоит в том, что каждым адресатом ИС генерируются два ключа, связанные между собой по определенному правилу. Один ключ объявляется открытым, а другой закрытым. Открытый ключ публикуется и доступен любому, кто желает послать сообщение адресату. Секретный ключ сохраняется в тайне.

Исходный текст шифруется открытым ключом адресата и передается ему. Зашифрованный текст в принципе не может быть расшифрован тем же открытым ключом. Дешифрование сообщение возможно только с использованием закрытого ключа, который известен только самому адресату.

Криптографические системы с открытым ключом используют так называемые необратимые или односторонние функции, которые обладают следующим свойством: при заданном значении x относительно просто вычислить значение $f(x)$, однако если $y=f(x)$, то нет простого пути для вычисления значения x.

Множество классов необратимых функций и порождает все разнообразие систем с открытым ключом. Однако не всякая необратимая функция годится для использования в реальных ИС. В самом определении необратимости присутствует неопределенность. Под *необратимостью* понимается не теоретическая необратимость, а практическая невозможность вычислить обратное значение, используя современные вычислительные средства за обозримый интервал времени.

Поэтому чтобы гарантировать надежную защиту информации, к системам с открытым ключом (СОК) предъявляются два важных и очевидных требования:

1. Преобразование исходного текста должно быть необратимым и исключать его восстановление на основе открытого ключа.
2. Определение закрытого ключа на основе открытого также должно быть невозможным на современном технологическом уровне. При этом желательна точная нижняя оценка сложности (количества операций) раскрытия шифра.

Алгоритмы шифрования с открытым ключом получили широкое распространение в современных информационных системах. Так, алгоритм RSA стал мировым стандартом де-факто для открытых систем.

Вообще же все предлагаемые сегодня криптосистемы с открытым ключом опираются на один из следующих типов необратимых преобразований: Разложение больших чисел на простые множители.

Вычисление логарифма в конечном поле. Вычисление корней алгебраических уравнений.

Здесь же следует отметить, что алгоритмы криптосистемы с открытым ключом (СОК) можно использовать в различных назначениях:
1. как самостоятельные средства защиты передаваемых и хранимых данных;
2. как средства для распределения ключей.

Алгоритмы СОК более трудоемки, чем традиционные криптосистемы. Поэтому часто на практике рационально с помощью СОК распределять ключи, объем которых как информации незначителен. А потом с помощью обычных симметричных алгоритмов осуществлять обмен большими информационными потоками.

Несмотря на довольно большое число различных СОК, наиболее популярна - криптосистема RSA, разработанная в 1977 году и получившая название в честь ее создателей: Рона Ривеста , Ади Шамира и Леонарда Эйдельмана. Они воспользовались тем фактом, что нахождение больших простых чисел в вычислительном отношении осуществляется легко, но разложение на множители произведения двух таких чисел практически невыполнимо.

Доказано (теорема Рабина), что раскрытие шифра RSA эквивалентно такому разложению. Поэтому для любой длины ключа можно дать нижнюю оценку числа операций для раскрытия шифра, а с учетом производительности современных компьютеров оценить и необходимое на это время. Возможность гарантированно оценить защищенность алгоритма RSA стала одной из причин популярности этой СОК на фоне десятков других схем.

В настоящее время алгоритм RSA широко используется в банковских компьютерных сетях, особенно для работы с удаленными клиентами (обслуживание кредитных карточек). Указанный алгоритм используется во многих стандартах, среди которых SSL, S-HHTP, S-MIME, S/WAN, STT и PCT.

Основными математическими результатами, положенными в основу этого алгоритма являются: малая теорема Ферма и функция Эйлера.

Открытый текст шифруется блоками, каждый из которых содержит двоичное значение, меньшее некоторого заданного числа *n*. Это значит, что длина блока должна быть меньше или равна $\log_2(n)$. На практике длина блока выбирается равной 2^k битам, где $2^k < n < 2^{k+1}$.

Шифрование и дешифрование для блока открытого текста *M* и блока шифрованного текста *C* можно представить в виде:

$$C = M^e \bmod n$$

$$M = C^d \bmod n = M^{ed} \bmod n = M \bmod n.$$

Отправитель и получатель должны знать значение *n*. Отправитель знает значение *e* и только получатель знает значение *d*.

Таким образом, открытым ключом является $\{e, n\}$, а личным закрытым ключом $\{d, n\}$.

При этом должны быть выполнены следующие требования:

- Должны существовать такие значения *e, d* и *n,* при которых выполняется $M^{ed} \bmod n = M \bmod n$ для всех значений $M <$ n.
- Должны относительно легко вычисляться M^e и C^d для всех значений $M <$ n.
- Должно быть практически невозможно определить *d* по имеющимся *e* и *n.*

Схема шифрования выглядит следующим образом:

1. Выбираются два простых числа *p* и *q* . (Например, *p=7 и q=17*).
2. Вычисляется $n = p \cdot q$. ($n = 119$).
3. Определяется $\varphi(n)=(p-1)(q-1)$. ($\varphi(n)=96$).
4. Выбор числа *e*, взаимно простого с $\varphi(n)$, причем $e < \varphi(n)$. ($e = 5$).
5. Вычисляется $d = e^{-1} \bmod \varphi(n)$.

   (Определяется такое *d*, что $d \cdot e = 1 \bmod 96$ и $e < 96$.
   Соответствующим значением будет $d = 77$, так как
   $77 \cdot 5 = 385 = 4 \cdot 96+1$).
6. Открытым ключом является $\{e, n\}$. ($\{e, n\}=\{5, 119\}$).
7. Закрытым ключом является $\{d, n\}$. ($\{d, n\}=\{77, 119\}$).
8. Шифрование $C = M^e \bmod n$
9. Дешифрование $M = C^d \bmod n$

Рассмотрим небольшой пример, иллюстрирующий применение алгоритма RSA.

Пример

Зашифруем сообщение "CAB".

Для простоты будем использовать маленькие простые числа (на практике применяются гораздо большие). Выберем $p = 3$ и $q = 11$. Определим $n = 3 \cdot 11 = 33$. Найдем $(p-1)(q-1) = 20$.

Следовательно, в качестве *e*, взаимно просто с $\varphi(n)=20$, например, $e = 7$. Выберем число *d*. В качестве такого числа может быть взято любое число, для которого удовлетворяется соотношение $(d \cdot e)(\text{mod}20) = (d \cdot 7)(\text{mod}20) = 1$, например $d = 3$.

Представим шифруемое сообщение как последовательность целых чисел с помощью отображения: $A \to 1, B \to 2, C \to 3$. Тогда сообщение принимает вид (3,1,2).

Зашифруем сообщение с помощью ключа $\{7,33\}$:

$(3^7)(\text{mod}33) = 2187(\text{mod}33) = 9,$

$(1^7)(\text{mod}33) = 1(\text{mod}33) = 1,$

$(2^7)(\text{mod}33) = 128(\text{mod}33) = 29.$

Расшифруем полученное зашифрованное сообщение (9,1,29) на основе закрытого ключа {3,33}:

$(9^3)(\mathrm{mod}33) = 729(\mathrm{mod}33) = 3,$

$(1^3)(\mathrm{mod}33) = 1(\mathrm{mod}33) = 1,$

$(29^3)(\mathrm{mod}33) = 24389(\mathrm{mod}33) = 2.$

Итак, в реальных системах алгоритм RSA реализуется следующим образом: каждый пользователь выбирает два больших простых числа, и в соответствии с описанным выше алгоритмом выбирает два простых числа e и d. Как результат умножения первых двух чисел (p,q) устанавливается n, $\{e,n\}$ образует открытый ключ, а $\{d,n\}$ - закрытый (хотя можно взять и наоборот).

Открытый ключ публикуется и доступен каждому, кто желает послать владельцу ключа сообщение, которое зашифровывается указанным алгоритмом. После шифрования, сообщение невозможно раскрыть с помощью открытого ключа. Владелец же закрытого ключа без труда может расшифровать принятое сообщение.

Задание

1. Создать программную реализацию алгоритма RSA.
2. Зашифровать введенное с клавиатуры сообщение, используя открытый ключ.
3. Расшифровать сообщение, используя закрытый ключ.

7. Комплекс криптоалгоритмов PGP

Общие сведения

Комплекс криптоалгоритмов PRETTY GOOD PRIVACY (PGP, весьма хорошая секретность) – программа для ведения секретной переписки, ставшая стандартом де-факто в гражданской криптографии. PGP представляет собой комплект программ, позволяющий шифровать и подписывать электронные сообщения. Реализации PGP имеются для большинства операционных систем:

- Windows (95,NT,2000);
- UNIX (Linux, FreeBSD, и др.);
- MAC;
- MS-DOS;
- BeOS;
- VAX/VMS и других.

PGP умеет встраиваться в популярные почтовые программы для Windows:

- Outlook,

- OutlookExpress,
- Eudora,

что делает его использование достаточно легким для пользователя. Кроме того, поддержка PGP имеется в российской программе для работы с почтой TheBat.

PGP – это свободно распространяемая программа безопасной электронной почты. Разработана Филипом Циммерманном (Philip Zimmermann), а выпущена фирмой Phil's Pretty Good Software [6]. PGP является криптографической системой с высокой степенью секретности. PGP позволяет пользователям обмениваться файлами или сообщениями:

1. с использованием функций секретности;
2. с установлением подлинности;
3. с высокой степенью удобства.

Секретность предполагает прочтение сообщения только адресатом. *Установление подлинности* позволяет определить, что сообщение, полученное от какого-либо человека, было послано именно им, а не кем-нибудь другим. Нет необходимости использовать специальные секретные каналы связи, что делает PGP простым в использовании программным обеспечением. Это связано с тем, что PGP базируется на мощной новой технологии, которая называется *шифрованием с "общим ключом"*.

PGP объединяет в себе удобство использования криптографической системы с общим ключом RSA и скорость обычной криптографической системы, алгоритм "дайджеста сообщений" для реализации электронной подписи, упаковку данных перед шифрованием, хороший эргономический дизайн программы и развитую систему управления ключами. PGP выполняет функции общего ключа быстрее, чем большинство других аналогичных реализаций этого алгоритма.

Как уже было упомянуто, шифрование сообщений и электронная подпись реализована в PGP на основе технологии шифрования с открытым ключом. Для пользователя это значит, что у него (и у каждого, кто пользуется системой) имеется 2 ключа: открытый (или публикуемый) – public key, закрытый (или частный) – private key.

Публикуемый ключ пользователь раздает тем, с кем ведет шифрованную переписку. Этот ключ можно публиковать где угодно – он не содержит пароля. Тот, кто будет писать шифрованное письмо, должен иметь публикуемый ключ своего адресата. Сам ключ представляет собой текстовый файл не очень понятного содержания.

Частный ключ – это ключ, который пользователь хранит у себя и никому не показывает. Только имея свой частный ключ, пользователь расшифровывает электронные письма и убеждается в подлинности подписи.

Отличительная черта технологии шифрования с открытым ключом состоит в том, что пользователь не говорит свой пароль никому (ведь

сам пароль тоже можно украсть, перехватить и т.д.), но при этом пользуется им для шифрованной переписки.

PGP – это не единственная система шифрования корреспонденции (известна также, например, система VeriSign). Однако технология PGP имеет несколько неоспоримых ***преимуществ.***

- Технология имеется для большинства операционных систем (многоплатформенность).
- Комплект программ является бесплатным и свободно распространяется для всех операционных систем.
- Технология не привязана к какому-либо центральному серверу. Открытые ключи можно передавать как угодно.
- Использовать PGP очень просто, поскольку он встраивается в почтовые программы.

Шифрование с общим ключом

В стандартных криптографических системах, таких как US Federal Data Encryption Standard (DES), один и тот же ключ используется для шифрования и расшифровки. Это значит, что ключ должен первоначально быть передан через секретные каналы так, чтобы обе стороны могли иметь его до того, как шифрованные сообщения будут посылаться по обычным каналам. Это может быть неудобно. Если имеется секретный канал для обмена ключами, тогда зачем нужна криптография?

В криптографической системе с общим ключом каждый имеет два связанных ключа: публикуемый общий ключ и секретный ключ. Каждый из них дешифрует код, сделанный с помощью другого. Знание общего ключа не позволяет вычислить соответствующий секретный ключ. Общий ключ может публиковаться и широко распространяться через коммуникационные сети. Такой протокол обеспечивает секретность без необходимости использовать специальные каналы связи, необходимые для стандартных криптографических систем.

Кто угодно может использовать общий ключ получателя, чтобы зашифровать сообщение ему, а получатель использует его собственный соответствующий секретный ключ для расшифровки сообщения. Никто, кроме получателя, не может расшифровать его, потому что никто больше не имеет доступа к секретному ключу. Даже тот, кто шифровал сообщение, не будет иметь возможности расшифровать его.

Кроме того, обеспечивается также установление подлинности сообщения. Собственный секретный ключ отправителя может быть использован для шифрования сообщения, таким образом "подписывая" его. Так создается электронная подпись сообщения, которую получатель (или кто-либо еще) может проверять, используя общий ключ отправителя для расшифровки. Это доказывает, что отправителем был создатель сообщения, и что сообщение впоследствии не изменялось кем-либо, так как отправитель – единственный, кто обладает секретным ключом, с помощью

которого была создана подпись. Подделка подписанного сообщения невозможна, и отправитель не может впоследствии изменить свою подпись.

Эти два процесса могут быть объединены для обеспечения и секретности, и установления подлинности: сначала подписывается сообщение собственным секретным ключом отправителя, а потом шифруется уже подписанное сообщение общим ключом получателя. Получатель делает наоборот: расшифровывает сообщение с помощью собственного секретного ключа, а затем проверяет подпись с помощью общего ключа отправителя сообщения. Эти шаги выполняются автоматически с помощью программного обеспечения получателя.

В связи с тем что алгоритм шифрования с общим ключом значительно медленнее, чем стандартное шифрование с одним ключом, шифрование сообщения лучше выполнять с использованием быстрого высококачественного стандартного алгоритма шифрования с одним ключом. Первоначальное незашифрованное сообщение называется *"открытым текстом"* (или просто текстом). В процессе, невидимом для пользователя, временный произвольный ключ, созданный только для этого одного "сеанса", используется для традиционного шифрования файла открытого текста. Тогда общий ключ получателя используется только для шифровки этого временного произвольного стандартного ключа. Зашифрованный ключ "сеанса" посылается, наряду с зашифрованным текстом (называемым ciphertext – зашифрованный), получателю. Получатель использует свой собственный секретный ключ, чтобы восстановить этот временный ключ сеанса, и затем применяет его для выполнения быстрого стандартного алгоритма декодирования с одним ключом, чтобы декодировать все зашифрованное сообщение [3, 6].

Сертификаты ключей

Общие ключи хранятся в виде "сертификатов ключей", которые включают в себя:

- идентификатор пользователя владельца ключа (обычно это имя пользователя);
- временную метку, которая указывает на время генерации пары ключей;
- собственно ключи.

Сертификаты общих ключей содержат общие ключи, а сертификаты секретных ключей – секретные. Каждый секретный ключ также шифруется с отдельным паролем. Файл ключей содержит один или несколько таких сертификатов. В каталогах общих ключей хранятся сертификаты общих ключей, а в каталогах секретных – сертификаты секретных ключей.

На ключи также внутренне ссылаются "идентификаторы ключей", которые являются "сокращением" общего ключа (самые младшие 64 бита

большого общего ключа). Когда этот идентификатор ключа отображается, то показываются лишь младшие 24 бита для краткости. Если несколько ключей могут одновременно использовать один и тот же идентификатор пользователя, то никакие два ключа не могут использовать один и тот же идентификатор ключа.

Дайджесты сообщений

PGP использует "дайджесты сообщений" для формирования подписи. **Дайджест сообщения** – это криптографически мощная 128-битная односторонняя хэш-функция от сообщения. Она несколько напоминает контрольную сумму, или CRC-код. Она однозначно представляет сообщение и может использоваться для обнаружения изменений в сообщении. В отличие от CRC-кода (контроля циклическим избыточным кодом), дайджест не позволяет создать два сообщения с одинаковым дайджестом. Дайджест сообщения шифруется секретным ключом для создания электронной подписи сообщения.

Документы подписываются посредством добавления перед ними удостоверяющей подписи, которая содержит идентификатор ключа, использованного для подписи; подписанный секретным ключом дайджест сообщения и метку даты и времени, когда подпись была сгенерирована. Идентификатор ключа используется получателем сообщения, чтобы найти общий ключ для проверки подписи. Программное обеспечение получателя автоматически ищет общий ключ отправителя и идентификатор пользователя в каталоге общих ключей получателя.

Шифрованным файлам предшествует идентификатор общего ключа, который был использован для их шифрования. Получатель использует этот идентификатор для поиска секретного ключа, необходимого для расшифровки сообщения. Программное обеспечение получателя автоматически ищет требуемый для расшифровки секретный ключ в каталоге секретных ключей получателя.

Эти два типа каталогов ключей и есть главный метод сохранения и управления общими и секретными ключами. Вместо того чтобы хранить индивидуальные ключи в отдельных файлах ключей, они собираются в каталогах ключей для облегчения автоматического поиска ключей либо по идентификатору ключа, либо по идентификатору пользователя. Каждый пользователь хранит свою собственную пару каталогов ключей. Индивидуальный общий ключ временно хранится в достаточно большом отдельном файле.

Алгоритмы секретных ключей

PGP предлагает на выбор различные алгоритмы секретных ключей. Под *алгоритмом секретного ключа* понимается симметричный блочный шифр,

использующий один ключ для шифрования и дешифрирования. Могут быть использованы следующие три симметричных блочных шифра:

- CAST,
- Triple-DES,
- IDEA.

На рис. 2 приведен пример диалоговой панели для установления свойств PGP.

Все три шифра работают с 64-битовыми блоками открытого текста и шифротекста. Длина ключа для CAST и IDEA – 128 битов, в то время как Triple-DES использует 168-битовый ключ. Подобно Data Encryption Standard (DES), любой из этих шифров может быть использован в режимах CFB и CBC. Криптографический **режим** обычно объединяет базовый шифр, какую-то обратную связь и ряд простых операций. FIPS PUB 81 определяет четыре режима работы: ECB, CBC, OFB и CFB. Банковские стандарты ANSI определяют для шифрования ECB и CBC, а для проверки подлинности – CBC и n-битовый CFB.

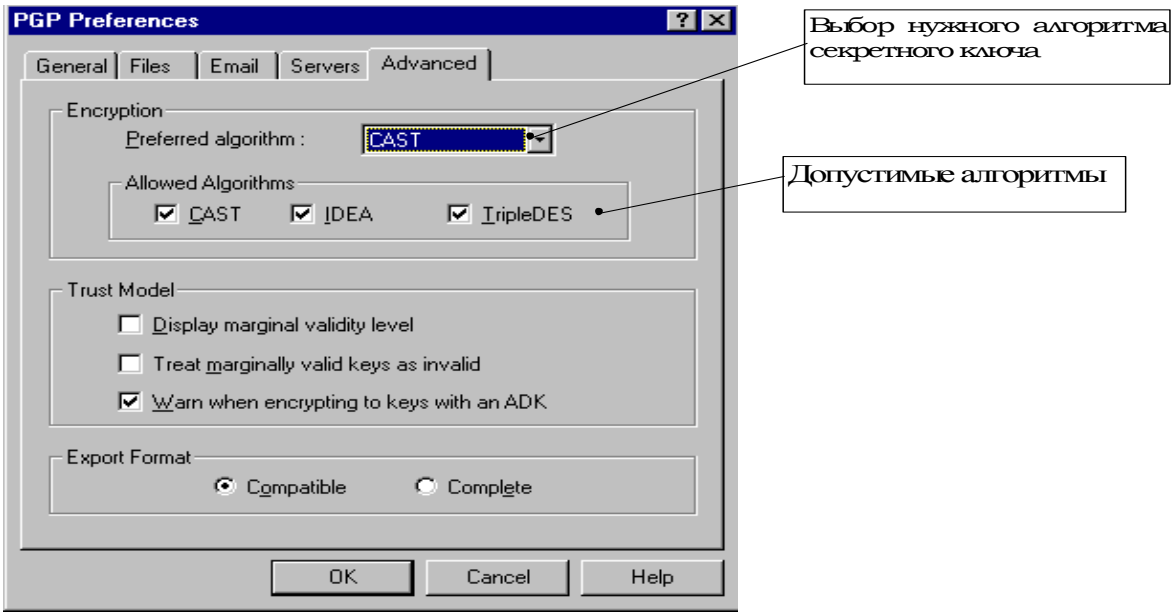

Рис. 2. Пример диалоговой панели для установления свойств PGP

Задание

1. Установить на 2 компьютера свободно распространяемую версию PGP.
2. Встроить в почтовую программу комплекс PGP.
3. Осуществить распределение ключей и обмен зашифрованными сообщениями между 2 абонентами криптосистемы PGP.

4. Применить функцию создания защищенного логического диска на компьютере пользователя

Основная литература

1. Смарт Н. Криптография. - Учебник для вузов: / пер. С. А. Кулешов, ред. пер. С. К. Ландо. - М. : Техносфера, 2005. - 528 с. – 11 экз.
2. Зубов А. Ю. Криптографические методы защиты информации. Совершенные шифры : Учебное пособие для вузов / А. Ю. Зубов. - М.: "Гелиос АРВ", 2005. – 190 с. – 30 экз.

Дополнительная литература

3. Молдовян А.А., Молдовян Н.А., Советов Б.Я. Криптография. - Серия «Учебники для вузов. Специальная литература». - СПб.: Издательство «Лань», 2000, 224 с. – 6 экз.
4. Петров А. А. Компьютерная безопасность. Криптографические методы защиты. – М.: ДМК, 2000, 448 с. - 3 экз.
5. Мельников В.П., Клейменов С.А., Петраков А.М.; / ред.: С. А. Клейменов Информационная безопасность и защита информации. Учебное пособие для вузов. - М. : Academia, 2006. - 330, 158 с. – 30 экз.
6. Шнайер Б. Прикладная криптография: Протоколы, алгоритмы, исходные тексты на языке Си : Пер. с англ. / Брюс Шнайер; Ред. пер. П. В. Семьянов. - М. : ТРИУМФ, 2003. 796 с. – 1 экз.
7. Завгородний В.И. Комплексная защита информации в компьютерных системах: Учебное пособие для вузов - М. : Логос, 2001. - 264 с. – 10 экз.
8. Романец Ю.В. Защита информации в компьютерных системах и сетях / Ю. В. Романец, П. А. Тимофеев, В. Ф. Шаньгин ; ред. : В. Ф. Шаньгин. - 2-е изд., перераб и доп. - М. : Радио и связь, 2001. - 376 с. – 42 экз.
9. http://www.ssl.stu.neva.ru/- Санкт-Петербургский центр защиты информации