Задания на контрольную работу

**ФИО : Зиборова Надежда Владимировна**

**Задание №3**. Алгоритм шифрования RSA.

Сгенерируйте открытый и закрытый ключи в алгоритме шифрования RSA, выбрав простые числа p и q из первой сотни. Зашифруйте сообщение, состоящее из ваших инициалов: ФИО.

**Задание №4**. Функция хеширования.

Найти хеш–образ своей Фамилии, используя хеш– функцию , где n = pq, p, q взять из Задания №3.

**Задание №5.** Электронная цифровая подпись.

Используя хеш-образ своей Фамилии, вычислите электронную цифровую подпись по схеме RSA.

**Пример**

**Задание №3**. Алгоритм шифрования RSA. Сгенерируйте открытый и закрытый ключи в алгоритме шифрования RSA, выбрав простые числа p и q из первой сотни. Зашифруйте сообщение, состоящее из ваших инициалов:

ФИО.

**I.Генерация ключей** (см. Приложение Г).

Выберем два простых числа р = 13 и q = 19 (см. Приложение Д).

Тогда модуль

n = pq=13\*19 = 247 и функция Эйлера j(n) = (p-1)(q-1) = 12\*18 = 216.

Закрытый ключ d выбираем из условий d < j(n) и d взаимно просто с j(n),

т.е. d и j(n) не имеют общих делителей.

Пусть d = 25.

Открытый ключ e выбираем из условий e<j(n) и de=1(mod j(n)): e<216,

25e=1(mod 216).

Последнее условие означает, что число 25e-1 должно делиться на 216 без остатка.

Таким образом, для определения e нужно подобрать такое число k, что

25e-1 = 216 k.

При k=14 получаем 25e=3024+1 или e=121.

В нашем примере (121, 247) – открытый ключ, ( 25, 247) – секретный ключ.

**II. Шифрование.**

Представим шифруемое сообщение «КГЛ» как последова-тельность целых

чисел. Пусть буква «К» соответствует числу 12, буква «Г» - числу 4 и буква

«Л» - числу 13.

Зашифруем сообщение, используя открытый ключ (121, 247):

С1 = (12 121 ) mod 247= 12

С2 = (4 121 ) mod 247=199

С3 = ( 13 121  ) mod 247= 91

Таким образом, исходному сообщению (12, 4, 13) соответствует

криптограмма (12, 199, 91).

III. Расшифрование

Расшифруем сообщение (12, 199, 91), пользуясь секретным ключом (25,247):

М1 = (12 25 ) mod 247=12

М2 = (199 25 ) mod 247= 4

МЗ = (91 25 ) mod 247=13

В результате расшифрования было получено исходное сообщение (12, 4, 13), то есть "КГЛ".

Замечания.





**Задание №4.** Функция хеширования. Найти хеш–образ своей Фамилии,

используя хеш–функцию , где n = pq, p, q взять из

Задания №3.

Хешируемое сообщение «КОЗИНА». Возьмем два простых числа p=13, q=19

(см. Приложение Е). Определим n=pq=13\*19=247. Вектор

инициализации выберем равным 8 (выбираем случайным образом). Слово

«КОЗИНА» можно представить последователь-ностью чисел (12, 16, 9, 10,

15, 1) по номерам букв в алфавите. Таким образом,

n=247, H0=8, M1=12, M2=16, M3=9, M4=10, M5=15, M6=1.

Используя формулу

,

получим хеш-образ сообщения «КОЗИНА»:

H1=(H0+M1)2 mod n = ( 8 + 12)2 mod 247 = 400 mod 247=153

H2=(H1+M2)2 mod n = (153 + 16)2 mod 247 = 28561 mod 247= 156

H3=(H2+M3)2 mod n = (156 + 9)2 mod 247 = 27225 mod 247= 55

H4=(H3+M4)2 mod n = ( 55 + 10)2 mod 247 = 4225 mod 247= 26

H5=(H4+M5)2 mod n = ( 26 + 15)2 mod 247 = 1681 mod 247= 199

H6=(H5+M6)2 mod n = (199 + 1)2 mod 247 = 40000 mod 247= 233

В итоге получаем хеш-образ сообщения «КОЗИНА», равный 233.

**Задание №5.** Электронная цифровая подпись. Используя хеш-образ своей Фамилии, вычислите электронную цифровую подпись по схеме RSA.

Пусть хеш-образ Фамилии равен 233, а закрытый ключ алгоритма RSA равен

(25, 247). Тогда электронная цифровая подпись сообщения, состоящего из

Фамилии, вычисляется по правилу (см. Приложение Ж)

s = 233 25 mod 247 = 168.

Для проверки ЭЦП, используя открытый ключ (121, 247), найдем

H = 168 121 mod 247 = 233.

Поскольку хеш-образ сообщения совпадает с найденным значением H, то

подпись признается подлинной.

**Приложение А. Алгоритм шифрования ГОСТ 28147-89**

Межгосударственный стандарт шифрования ГОСТ 28147-89

предусматривает 4 режима работы:

· режим простой замены;

· режим гаммирования;

· режим гаммирования с обратной связью;

· режим выработки имитовставки.

**Простая замена.**

Режим простой замены является основой для всех остальных режимов. Длина блока - 64 бита, длина ключа – 256 бит, количество подключей – 32, длина подключа - 32 бита, число циклов –32.

Открытые данные, подлежащие зашифрованию, разбиваются на 64-битные блоки, которые обрабатываются независимо друг от друга (Так как блоки данных шифруются независимо друг от друга, при зашифровании двух одинаковых блоков открытого текста получаются одинаковые блоки шифротекста и наоборот.). Схема обработки 64-битного блока показана на

Рис.1-2.

Процедура зашифрования 64-битного блока включает 32 цикла. В каждом цикле используется свой подключ, который вырабатывается из основного ключа. Размер массива открытых или зашифрованных данных,

подвергающийся соответственно зашифрованию или расшифрованию, должен быть кратен 64 битам, после выполнения операции размер полученного массива данных не изменяется.

Режим простой замены применяется для шифрования короткой, ключевой информации.

В режимах гаммирования вырабатывается гамма шифра блоками по 64 бита с применением ГОСТ в режиме простой замены. В первом режиме гамма не зависит от шифруемых данных, во втором – зависит от шифрблоков.

Режим выработки имитовставки предназначен для обнаружения случайных или умышленных искажений данных. Имитовставка вырабатывается (с помощью первых 16 циклов ГОСТ в режиме простой замены) из открытых данных и ключа и добавляется при передаче по каналу связи к блокам

зашифрованных данных.



где Å - сложение по модулю 2

Рис. 1. Алгоритм шифрования ГОСТ 28147-89 (режим простой замены).



Рис. 2. Функция преобразования f(R,X) в алгоритме ГОСТ 28147-89.

**Приложение Б. Символы кириллицы (альтернативная кодовая таблица ASCII)**









**Приложение В. Блок подстановки в алгоритме шифрования ГОСТ**







Приложение Г. Алгоритм шифрования RSA

Алгоритм шифрования RSA относится к криптографическим системам с

открытым ключом. Криптосистемы с открытым ключом (асимметричные

криптосистемы) были разработаны во второй половине семидесятых годов. В асимметричных криптосистемах процедуры прямого и обратного криптопреобразования выполняются на различных ключах и не имеют между собой очевидных и легко прослеживаемых связей, позволяющих по одному ключу определить другой. В такой схеме знание только ключа зашифрования не позволяет расшифровать сообщение, поэтому он не является секретным элементом шифра и обычно публикуется участником обмена для того, чтобы любой желающий мог послать ему шифрованное сообщение.

Принцип функционирования асимметричной криптосистемы заключается в

следующем:

· пользователь А генерирует два ключа - открытый (незасекречен-ный) и

секретный - и передает открытый ключ по незащищенному каналу

пользователю Б;

· пользователь Б шифрует сообщение, используя открытый ключ

шифрования пользователя А;

· пользователь Б посылает зашифрованное сообщение пользователю А

по незащищенному каналу;

· пользователь А получает зашифрованное сообщение и дешифрует его, используя свой секретный ключ.

Пары {открытый ключ; секретный ключ} вычисляются с помощью специальных алгоритмов, причем ни один ключ не может быть выведен из

другого.

Криптографическая система RSA (Rivest-Shamir-Adleman)

Авторами алгоритма RSA, предложенного в 1977 г., являются Р.Риверст (Rivest), А.Шамир (Shamir) и А.Адлеман (Adleman). Надежность алгоритма основывается на трудности факторизации (разложения на множители) больших чисел и трудности вычисления дискретных алгоритмов (нахождения x при известных a, b и n из уравнения a

x

= b (mod n) ).

Алгоритм RSA состоит из трех частей: генерации ключей, шифрования и

расшифрования.

1. Генерация ключей.

Выберем два больших различных простых числа p и q (Натуральное число называется простым, если оно делится только на себя и на 1.) и найдем их произведение

n = pq .

Вычислим функцию Эйлера j(n) по формуле

j(n) = (p-1)(q-1).

Закрытый ключ d выбираем из условий d < j(n) и d взаимно просто с j(n),

т.е. d и j(n) не имеют общих делителей.

Открытый ключ e выбираем из условий e < j(n) и de = 1(mod j(n)) .

Последнее условие означает, что разность de - 1 должна делить-ся на j(n) без остатка. Для определения числа e нужно подобрать такое число k, что

de - 1 = j(n)\*k .

В алгоритме RSA

( e, n ) – открытый ключ,

( d, n ) – секретный ключ.

2. Шифрование.

Исходное сообщение разбивается на блоки Mi одинаковой длины. Каждый блок представляется в виде большого десятичного числа, меньшего n, и шифруется отдельно. Шифрование блока M (M - десятичное число) осуществляется по следующей формуле

Me= C (mod n) , где C – шифрблок, соответствующий блоку открытого сообщения M. Шифрблоки соединяются в шифрограмму.

3. Расшифрование.

При расшифровании шифрограмма разбивается на блоки известной длины и каждый шифрблок расшифровывается отдельно по следующей формуле

Cd= M (mod n) .





Приложение Е. Функция хеширования

Функцией хеширования (хеш-функцией) называется преобразо-вание данных, переводящее строку битов M произвольной длины в строку битов h(M) некоторой фиксированной длины (несколько десятков или сотен бит).

Хеш-функция h(M) должна удовлетворять следующим условиям:

1. хеш-функция h(M) должна быть чувствительна к любым изменениям

входной последовательности M;

2. для данного значения h(M) должно быть невозможно найти

значение M;

3. для данного значения h(M) должно быть невозможно найти

значение M’

¹ M такое, что h(M‘) = h(M) .

Ситуация, при которой для различных входных последова-

тельностей M , M‘ совпадают значения их хеш-образов:

 h(M) = h(M‘), называется коллизией.

При построении хеш-образа входная последовательность M разбивается на блоки Mi фиксированной длины и обрабатывается поблочно по

формуле H i = f(H i-1 , M i ).

Хеш-значение, вычисляемое при вводе последнего блока сообщения, становится хеш-значением (хеш-образом) всего сообщения.

В качестве примера рассмотрим упрощенный вариант хеш-функции из рекомендаций МККТТ Х.509:

H i  = (H i-1 + M i ) 2 mod n ,

где n = pq, p и q – большие простые числа, H0 - произвольное начальное

заполнение, Mi - i-тый блок сообщения M = M1 M2 … Mk

**Приложение Ж. Электронная цифровая подпись**

Цифровая подпись в цифровых документах играет ту же роль, что и подпись, поставленная от руки в документах на бумаге: это данные, присоединяемые к передаваемому сообщению, подтвержда-ющие, что владелец подписи составил или заверил это сообщение. Получатель сообщения с помощью цифровой подписи может проверить, что автором сообщения является именно владелец подписи и что в процессе передачи не была нарушена целостность полученных данных.

При разработке механизма цифровой подписи возникают следующие задачи:

создать подпись таким образом, чтобы ее невозможно было подделать;

· иметь возможность проверки того, что подпись действительно

принадлежит указанному владельцу;

· иметь возможность предотвратить отказ от подписи.

Классическая схема создания цифровой подписи

При создании цифровой подписи по классической схеме отправитель

1. применяет к исходному сообщению хеш-функцию;

2. вычисляет цифровую подпись по хеш-образу сообщения с использованием секретного ключа создания подписи;

3. формирует новое сообщение, состоящее из исходного сообщения и добавленной к нему цифровой подписи.

Получатель, получив подписанное сообщение,

1. отделяет цифровую подпись от основного сообщения;

2. применяет к основному сообщению хеш-функцию;

3. с использованием открытого ключа проверки подписи извлекает хеш- образ сообщения из цифровой подписи;

4. проверяет соответствие вычисленного хеш-образа сообщения (п.2) и извлеченного из цифровой подписи. Если хеш-образы совпадают, то подпись признается подлинной.

**Схема подписи RSA**

Криптосистема с открытым ключом RSA может использоваться не только для шифрования, но и для построения схемы цифровой подписи.

Для создания подписи сообщения ^ M отправитель

1. вычисляет хеш-образ r = h(M) сообщения M с помощью некоторой хеш-функции;

2. зашифровывает полученный хеш-образ r на своем секретном ключе (d,n) , т.е. вычисляет значение s = r d mod n , которое и является подписью.

Для проверки подписи получатель

1. расшифровывает подпись s на открытом ключе (e,n) отправителя, т.е. вычисляет r’ = s e mod n и таким образом восстанавливает предполагаемый хеш-образ r’сообщения M;

2. вычисляет хеш-образ h(M) = r сообщения M с помощью той же самой хеш-функции, которую использовал отправитель;

3. сравнивает полученные значения r и r’. Если они совпадают, то

подпись правильная, отправитель действительно является тем, за кого себя выдает, и сообщение не было изменено при передаче.