**Индивидуальные домашние задания №1**

**Задание №1. Шифр Цезаря**. Используя шифр Цезаря, зашифруйте свои данные: Фамилию Имя Отчество.

Исходный текст:

**«**КОЩАВЦЕВ ДАНИИЛ ВАСИЛЬЕВИЧ**»**

Используем алфавит, содержащий 33 буквы и пробел, стоящий после буквы Я:

АБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ*пробел*

Ключом в шифре Цезаря является число 3. Каждая буква в исходном тексте сдвигается по алфавиту на 3 позиции. Таким образом, получаем:

|  |  |
| --- | --- |
| Исходный текст | КОЩАВЦЕВ ДАНИИЛ ВАСИЛЬЕВИЧ |
| Зашифрованный текст | НСЬГЕЩЗЕВЖГРЛЛОВЕГФЛОЯЗЕЛЪВ |

**Задание №2. Алгоритм шифрования ГОСТ 28147-89.** Выполните первый цикл алгоритма шифрования ГОСТ 28147-89 в режиме простой замены (см. Приложение А). Для получения 64 бит исходного текста *используйте 8 первых символов из своих данных*: Фамилии Имени Отчества. Для получения ключа (256 бит) используют текст, состоящий из 32 букв. *Первый подключ содержит первые 4 буквы.*

Исходные данные для зашифрования из 32 букв: КОЩАВЦЕВ ДАНиил Васильевич и криптография.

В качестве первого блока используем 8 первых символов: КОЩАВЦЕВ Для первого подключа Х используем первые 4 буквы ключа: ДАН.

Переводим исходный текст и первый подключ в двоичную последовательность (см. Приложение Б):

К – 11001010

О – 11001110

Щ – 11011001

А – 11000000

В – 11000010

Ц – 11010110

Е – 11000101

В – 11000010

первый подключ X0

Пробел – 00100000

Д – 11000100

А – 11000000

Н – 11001101

Таким образом, первые 64 бита определяют входную последовательность

L0: 11001010 11001110 11011001 11000000

R0: 11000010 11010110 11000101 11000010

следующие 32 бита определяют первый подключ

Х0: 00100000 11000100 11000000 11001101

I. Найдем значение функции преобразования f(R0,X0) (см. Приложение А)

1). Вычисление суммы R0 и X0 по mod 232

R0: 1100 0010 1101 0110 1100 0101 1100 0010

Х0: 0010 0000 1100 0100 1100 0000 1100 1101

1110 0011 1001 1011 1000 0110 1000 1111

2). Преобразование в блоке подстановки

Результат суммирования R0+X0 по mod 232

1110 0011 1001 1011 1000 0110 1000 1111

преобразуем в блоке подстановки (см. Приложение В). Для каждого 4-битного блока вычислим его адрес в таблице подстановки. Номер блока соответствует номеру столбца, десятичное значение блока соответствует номеру строки в таблице.

номера блоков

8 7 6 5 4 3 2 1

1110 0011 1001 1011 1000 0110 1000 1111

соответствующие номера строк в таблице подстановки

8 7 6 5 4 3 2 1

14 3 9 11 8 6 8 15

1000 0001 0110 1110 1110 0100 0010 0011

3). Циклический сдвиг результата п.2 на 11 бит влево

0111 0111 0010 0001 0001 1100 0000 1011

Таким образом, нашли значение функции f (R0,X0):

0111 0111 0010 0001 0001 1100 0000 1011

II. Вычисляем R1= f(R0,X0) L0.

Результат преобразования функции f(R0,X0) складываем с L0 по mod2:

L0: 1100 1010 1100 1110 1101 1001 1100 0000

f(R0,X0): 0111 0111 0010 0001 0001 1100 0000 1011

R1: 0011 0001 1110 1111 1110 0101 1100 1011

**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* = 31 (см. Приложение Д). Тогда модуль *n* = *pq*=13\*31 = 403 и функция Эйлера

*(n)* = *(p-1)(q-1)* = 12\*30= 360.

Закрытый ключ *d* выбираем из условий *d* < *(n)* и *d* взаимно просто с *(n)*, т.е. *d* и *(n)* не имеют общих делителей. Пусть *d* = 107.

Открытый ключ *e* выбираем из условий *e*<*(n)* и *de*=1*(mod* *(n)):* *e*<360, 107*e*=1(mod 360).

Последнее условие означает, что число 107*e*-1 должно делиться на 360 без остатка. Таким образом, для определения *e* нужно подобрать такое число *k*, что 107*e*-1 = 360 *k*.

Будем подбирать это число с помощью расширенного алгоритма Евклида: делим с остатком 360 на 107:

делении

𝑃0 = 3, 𝑃1 = 𝑞1, 𝑃𝑖 = 𝑞𝑖 ∙ 𝑃𝑖−1 + 𝑃𝑖−2 для 𝑖 ≥ 2

Контроль: 𝑃𝑠 = 𝑛

Тогда 𝑎−1 ≡ (−1)𝑠−1𝑃𝑠−1(𝑚𝑜𝑑 𝑛)

****

s = 6. Контроль: 𝑃𝑠 = 𝑛, то есть 𝑃6 = 360 − верно. Значит, 107−1 ≡ (−1)𝑠−1𝑃𝑠−1 = (−1)537 ≡ 323 (𝑚𝑜𝑑 216).

Таким образом, (323, 403) – открытый ключ, (107, 403) – секретный ключ.

Замечание. После формирования ключа промежуточные числа (*p*, *q*, (*n*)) рекомендуется уничтожить.

1. Зашифрование.

Представим шифруемое сообщение «КДВ» как последовательность целых чисел.

Пусть буква «К» соответствует числу 12, буква «Д» - числу 5 и буква «В» - числу 3. Зашифруем сообщение, используя открытый ключ (323, 403):

𝐶1 = (12323) 𝑚𝑜𝑑 403 = 363

𝐶2 = (5323) 𝑚𝑜𝑑 403 = 242

𝐶3 = (3323) 𝑚𝑜𝑑 403 = 321

Таким образом, исходному сообщению (12, 5, 3) соответствует криптограмма (363, 242, 321).

1. Расшифрование

Расшифруем сообщение (363, 242, 321), пользуясь секретным ключом (107,403):

𝑀1 = (363107) 𝑚𝑜𝑑 403 = 12

𝑀2 = (242107) 𝑚𝑜𝑑 403 = 5

𝑀3 = (321107) 𝑚𝑜𝑑 403 = 3

В результате расшифрования было получено исходное сообщение (12, 5, 3), то есть "КДВ".

**Задание №4. Функция хеширования.** Найти хеш–образ своей Фамилии, используя хеш–функцию Hi =(Hi 1− +M modni )2 , где *n = pq*. Числа p, q взять из задания №3.

Хешируемое сообщение «КОЩАВЦЕВ». Возьмем два простых числа *p*=13, *q*=31 (см. Приложение Е). Определим *n*=*pq*=13\*19=247. Вектор инициализации H0 выберем равным

23 (выбираем случайным образом). Слово «САВИН» можно представить

последовательностью чисел (19, 1, 3, 10, 15) по номерам букв в алфавите. Таким образом, n=403, H0=38, M1=12, M2=16, M3=27, M4=1, M5=3, M6=24, M7=6, M8=3.

Используя формулу

Hi =(Hi 1− +M modni )2 ,

получим хеш-образ сообщения «КОЩАВЦЕВ»:

H1 = mod 403 = 82

H2 = mod 403 = 335

H3 = mod 403 = 69

H4 = mod 403 = 64

H5 = mod 403 = 56

H6 = mod 403 = 355

H7 = mod 403 = 152

H8 = mod 403 = 248

итоге получаем хеш-образ сообщения «КОЩАВЦЕВ», равный 248.

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

Пусть хеш-образ Фамилии равен 248, а закрытый ключ алгоритма RSA равен (107, 403). Тогда электронная цифровая подпись сообщения, состоящего из Фамилии, вычисляется по правилу (см. Приложение Ж) *s* = 248 107 mod 403 = 248.

Для проверки ЭЦП, используя открытый ключ (323, 403), найдем H = 248 323 mod 403 = 248.

Поскольку хеш-образ сообщения совпадает с найденным значением H, то подпись признается подлинной.

Замечание. Обратите внимание, что в данном случае хешированное сообщение совпало с исходным. Такое недопустимо в криптографии. Однако, при большом значении m это крайне маловероятно.

# 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.

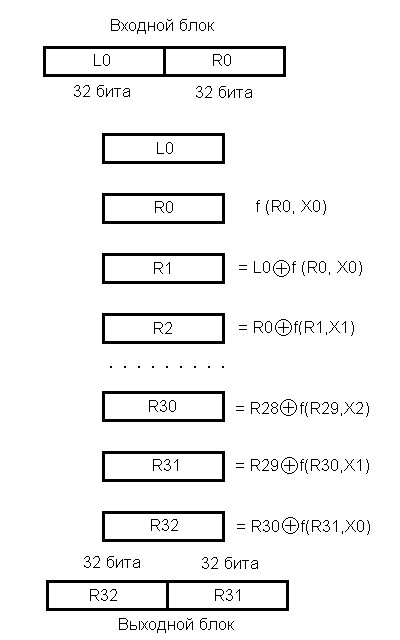
Открытые данные, подлежащие зашифрованию, разбиваются на 64-битные блоки, которые обрабатываются независимо друг от друга (Так как блоки данных шифруются независимо друг от друга, при зашифровании двух одинаковых блоков открытого текста получаются одинаковые блоки шифротекста и наоборот.). Схема обработки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

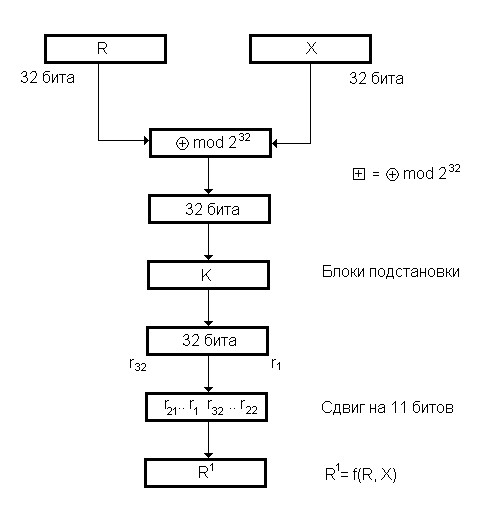
Режим простой замены применяется для шифрования короткой, ключевой информации.

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  - сложение по модулю 2

**Рисунок 1 - Алгоритм шифрования ГОСТ 28147-89 (режим простой замены)**



# Рисунок 2 - Функция преобразования *f(R,X)* в алгоритме ГОСТ 28147-89 Приложение Б. Символы кириллицы (альтернативная кодовая таблица ASCII)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Сим-л | Дес. | Двоич. | Сим-л | Дес. | Двоич. |
| А | 192 | 11000000 | б | 225 | 11100001 |
| Б | 193 | 11000001 | в | 226 | 11100010 |
| В | 194 | 11000010 | г | 227 | 11100011 |
| Г | 195 | 11000011 | д | 228 | 11100100 |
| Д | 196 | 11000100 | е | 229 | 11100101 |
| Е | 197 | 11000101 | ж | 230 | 11100110 |
| Ж | 198 | 11000110 | з | 231 | 11100111 |
| З | 199 | 11000111 | и | 232 | 11101000 |
| И | 200 | 11001000 | й | 277 | 11101001 |
| Й | 201 | 11001001 | к | 234 | 11101010 |
| К | 202 | 11001010 | л | 235 | 11101011 |
| Л | 203 | 11001011 | м | 236 | 11101100 |
| М | 204 | 11001100 | н | 237 | 11101101 |
| Н | 205 | 11001101 | о | 238 | 11101110 |
| О | 206 | 11001110 | п | 239 | 11101111 |
| П | 207 | 11001111 | р | 240 | 11110000 |
| Р | 208 | 11010000 | с | 241 | 11110001 |
| С | 209 | 11010001 | т | 242 | 11110010 |
| Т | 210 | 11010010 | у | 243 | 11110011 |
| У | 211 | 11010011 | ф | 244 | 11110100 |
| Ф | 212 | 11010100 | х | 245 | 11110101 |
| Х | 213 | 11010101 | ц | 246 | 11110110 |
| Ц | 214 | 11010110 | ч | 247 | 11110111 |
| Ч | 215 | 11010111 | ш | 248 | 11111000 |
| Ш | 216 | 11011000 | щ | 249 | 11111001 |
| Щ | 217 | 11011001 | ъ | 250 | 11111010 |
| Ъ | 218 | 11011010 | ы | 251 | 11111011 |
| Ы | 219 | 11011011 | ь | 252 | 11111100 |
| Ь | 220 | 11011100 | э | 253 | 11111101 |
| Э | 221 | 11011101 | ю | 254 | 11111110 |
| Ю | 222 | 11011110 | я | 255 | 11111111 |
| Я | 223 | 11011111 | пробел | 32 | 00100000 |
| а | 224 | 11100000 |  |  |  |

**Приложение В. Блок подстановки в алгоритме шифрования ГОСТ 28147-89**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |
| 0 | 1 | 13 | 4 | 6 | 7 | 5 | 14 | 4 |
| 1 | 15 | 11 | 11 | 12 | 13 | 8 | 11 | 10 |
| 2 | 13 | 4 | 10 | 7 | 10 | 1 | 4 | 9 |
| 3 | 0 | 1 | 0 | 1 | 1 | 13 | 12 | 2 |
| 4 | 5 | 3 | 7 | 5 | 0 | 10 | 6 | 13 |
| 5 | 7 | 15 | 2 | 15 | 8 | 3 | 13 | 8 |
| 6 | 10 | 5 | 1 | 13 | 9 | 4 | 15 | 0 |
| 7 | 4 | 9 | 13 | 8 | 15 | 2 | 10 | 14 |
| 8 | 9 | 0 | 3 | 4 | 14 | 14 | 2 | 6 |
| 9 | 2 | 10 | 6 | 10 | 4 | 15 | 3 | 11 |
| 10 | 3 | 14 | 8 | 9 | 6 | 12 | 8 | 1 |
| 11 | 14 | 7 | 5 | 14 | 12 | 7 | 1 | 12 |
| 12 | 6 | 6 | 9 | 0 | 11 | 6 | 0 | 7 |
| 13 | 11 | 8 | 12 | 3 | 2 | 0 | 7 | 15 |
| 14 | 8 | 2 | 15 | 11 | 5 | 9 | 5 | 5 |
| 15 | 12 | 12 | 14 | 2 | 3 | 11 | 9 | 3 |

Пример. Пусть 32-битная последовательность имеет вид

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 100  1 | 101  1 | 110  0 | 010  1 | 111  0 | 010  0 | 000  0 | 100  1 |

Разобьем входную последовательность на 8 блоков по 4 бита. Шестой блок 1100 пропускаем через 6-ой узел подстановки по следующему правилу: преобразуем двоичное число 1100 к десятичному виду – 12. Заполнение 12-ой строки для 6-ого узла подстановки равно 9, что в двоичном виде есть 1001. Таким образом, 4-битный блок 1100 заменяется на 1001. Остальные блоки заменяются аналогично.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | номер узла |
| 1001 | 1011 | 1100 | 0101 | 1110 | 0100 | 0000 | 1001 | вход |
| 9 | 11 | 12 | 5 | 14 | 4 | 0 | 9 | адрес |
| 2 | 7 | 9 | 15 | 5 | 10 | 14 | 11 | заполнение |
| 0010 | 0111 | 1001 | 1111 | 0101 | 1010 | 1110 | 1011 | результат |

Выходная последовательность имеет вид

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 001  0 | 011  1 | 100  1 | 111  1 | 010  1 | 101  0 | 111  0 | 101  1 |

# 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 (незасекреченный) 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 *ax* = *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 *(n)* по формуле

*(n)* = *(p-1)(q-1).*

Закрытый ключ *d* выбираем из условий *d* < *(n)* и

*d* взаимно просто с *(n)*,

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

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

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

В алгоритме RSA

(*e, n* ) – открытый ключ, (*d, n* ) – секретный ключ.

1. 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.

1. Расшифрование.

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

*Cd* = *M (mod n)* .

**Приложение Д. Таблица простых чисел**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 5 | 7 |
| 11 | 13 | 17 | 19 | 23 |
| 29 | 31 | 37 | 41 | 43 |
| 47 | 53 | 59 | 61 | 67 |
| 71 | 73 | 79 | 83 | 89 |
| 97 | 101 | 103 | 107 | 109 |
| 113 | 127 | 131 | 137 | 139 |
| 149 | 151 | 157 | 163 | 167 |
| 173 | 179 | 181 | 191 | 193 |
| 197 | 199 | 211 | 223 | 227 |
| 229 | 233 | 239 | 241 | 251 |
| 257 | 263 | 269 | 271 | 277 |
| 281 | 283 | 293 | 307 | 311 |
| 313 | 317 | 331 | 337 | 347 |
| 349 | 353 | 359 | 367 | 373 |
| 379 | 383 | 389 | 397 | 401 |
| 409 | 419 | 421 | 431 | 433 |
| 439 | 443 | 449 | 457 | 461 |
| 463 | 467 | 479 | 487 | 491 |
| 499 | 503 | 509 | 521 | 523 |
| 541 | 547 | 557 | 563 | 569 |
| 571 | 577 | 587 | 593 | 599 |

# Приложение Е. Функция хеширования

Функцией хеширования (хеш-функцией) называется преобразование 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’*  *M* такое, что *h(M‘)* = *h(M)* .

Ситуация, при которой для различных входных последовательностей *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

*Hi* = *f(Hi-1 , Mi)*.

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

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

*Hi* = *(Hi-1 + Mi )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, подтверждающие, 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* = *rd mod n* , которое и является подписью.

Для проверки подписи получатель

1. расшифровывает подпись *s* на открытом ключе *(e,n)* отправителя, т.е. вычисляет *r’* = *se 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 хешфункции, 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.