Индивидуальные домашние задания №1

Задание №1. Шифр Цезаря.

Используя шифр Цезаря, зашифруйте свои данные: Фамилию Имя Отчество.

Задание №2. Алгоритм шифрования ГОСТ 28147-89.

Выполните первый цикл алгоритма шифрования ГОСТ 28147 89 в режиме простой замены. 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.

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 $H_i = \left(H_{i-1} + M_i\right)^2 \bmod n$, 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.

Примеры выполнения заданий

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

Исходный текст:
«САВИН ВЛАДИМИР НИКОЛАЕВИЧ»

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	САВИН ВЛАДИМИР НИКОЛАЕВИЧ
Зашифрованный текст	ФГЕЛРВЕОГЖЛПЛУВРЛНСОГЗЕЛЪ

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: САВИН ВЛАДИМир Николаевич и криптография.

В качестве первого блока используем 8 первых символов: САВИН ВЛ

Для первого подключа X используем первые 4 буквы ключа: АДИМ.

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

исходный текст

С	11010001
А	11000000
В	11000010
И	11001000
Н	11001101
пробел	00100000
В	11000010
Л	11001011

первый подключ X0

А	11000000
Д	11000100
И	11001000
М	11001100

Таким образом, первые 64 бита определяют входную последовательность

L0: 11010001 11000000 11000010 11001000

R0: 11001101 00100000 11000010 11001011

следующие 32 бита определяют первый подключ

X0: 11000000 11000100 11001000 11001100

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

1). Вычисление суммы R0 и X0 по mod 2^{32}

R0:	1100 1101	0010 0000	1100 0010	1100 1011
X0:	1100 0000	1100 0100	1100 1000	1100 1100
	1000 1101	1110 0101	1000 1011	1001 0111

2). Преобразование в блоке подстановки

Результат суммирования R0+X0 по mod 2^{32}

1000 1101 1110 0101 1000 1011 1001 0111

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. Таким образом, 5-тый блок (1011) заменяется заполнением 11-ой строки и пятого столбца в таблице подстановки (1110).

номера блоков							
8	7	6	5	4	3	2	1
1000	1101	1110	0101	1000	1011	1001	0111
соответствующие номера строк в таблице подстановки							
8	13	14	5	8	11	9	7
заполнение							
9	8	15	15	14	7	3	14
результат							
1001	1000	1111	1111	1110	0111	0011	1110

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

1111 1111 0011 1001 1111 0100 1100 0111

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

1111 1111 0011 1001 1111 0100 1100 0111

II. Вычисляем R1= f(R0,X0) ⊕L0.

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

L0:	1101	0001	1100	0000	1100	0010	1100	1000
f(R0,X0):	1111	1111	0011	1001	1111	0100	1100	0111
R1:	0010	1110	1111	1001	0011	0110	0000	1111

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 $p = 13$ и $q = 19$ (см. Приложение Д).

Тогда модуль

$n = pq = 13*19 = 247$

и функция Эйлера

$\varphi(n) = (p-1)(q-1) = 12*18 = 216$.

Закрытый ключ d выбираем из условий $d < \varphi(n)$ и d взаимно просто с $\varphi(n)$, т.е. d и $\varphi(n)$ не имеют общих делителей.

Пусть $d = 25$.

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

$25e=1 (\mathrm{mod}\ 216)$.

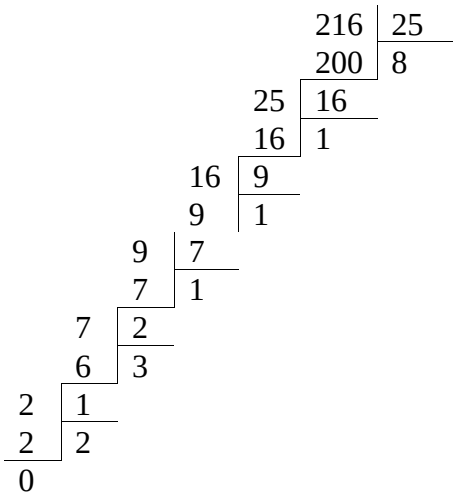
Последнее условие означает, что число $25e-1$ должно делиться на 216 без остатка.

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

$25e-1 = 216k$.

Будем подбирать это число с помощью расширенного алгоритма Евклида: делим с остатком 216 на 25:

$$\mathbf{216} = 8 \cdot \mathbf{25} + 16, \quad \mathbf{25} = 1 \cdot \mathbf{16} + 9, \quad \mathbf{16} = 1 \cdot \mathbf{9} + 7,$$
$$\mathbf{9} = 1 \cdot \mathbf{7} + 2, \quad \mathbf{7} = 3 \cdot \mathbf{2} + 1, \quad 2 = 2 \cdot 1$$



Последний ненулевой остаток в схеме Евклида – это НОД этих чисел. Выражаем теперь последовательно НОД, начиная с конца:

$$1 = \mathbf{7} - 3 \cdot \mathbf{2} = \mathbf{7} - 3 \cdot (\mathbf{9} - 1 \cdot \mathbf{7}) = -3 \cdot \mathbf{9} + 4 \cdot \mathbf{7} =$$
$$= -3 \cdot \mathbf{9} + 4 \cdot (\mathbf{16} - 1 \cdot \mathbf{9}) = 4 \cdot \mathbf{16} - 7 \cdot \mathbf{9} =$$
$$= 4 \cdot \mathbf{16} - 7 \cdot (\mathbf{25} - 1 \cdot \mathbf{16}) = -7 \cdot 25 + 11 \cdot 16 =$$
$$= -7 \cdot 25 + 11 \cdot (\mathbf{216} - 8 \cdot \mathbf{25}) = 11 \cdot 216 - 95 \cdot \mathbf{25}$$

По модулю 216 получим $-95 \cdot \mathbf{25} = 1 (mod\ 216)$, значит, $e = -95 \equiv -95 + 216 = 121\ (mod\ 216)$

Другой способ (сокращённый): выписать в таблицу полученные в схеме Евклида неполные частные q_i при делении a на n. Далее находим

$P_0 = 1, \quad P_1 = q_1, \quad P_i = q_i \cdot P_{i-1} + P_{i-2}$ для $i \geq 2$

Контроль: $P_s = n$

Тогда $a^{-1} \equiv (-1)^{s-1} P_{s-1} (mod\ n)$

В нашем примере

i	0	1	2	3	4	5	6
q_i	–	8	1	1	1	3	2
P_i	1	8	(8·1+1) 9	(1·9 + 8) 17	(1·17+9) 26	(3·26+17) 95	(2·95+26) 216

s = 6. Контроль: $P_s = n$, то есть $P_6 = 216$ – верно.

Значит, $25^{-1} \equiv (-1)^{s-1} P_{s-1} = (-1)^5 95 \equiv 121\ (mod\ 216)$.

Таким образом, (121, 247) – открытый ключ, (25, 247) – секретный ключ.

Замечание. После формирования ключа промежуточные числа (p, q, $\varphi(n)$) рекомендуется уничтожить.

II. Зашифрование.

Представим шифруемое сообщение «СВН» как последовательность целых чисел. Пусть буква «С» соответствует числу 19, буква «В» - числу 3 и буква «Н» - числу 15.

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

$$C_1 = (19^{121})\ mod\ 247 = 19$$
$$C_2 = (3^{121})\ mod\ 247 = 185$$
$$C_3 = (15^{121})\ mod\ 247 = 67$$

Таким образом, исходному сообщению (19, 3, 15) соответствует криптограмма (19, 185, 67).

III. Расшифрование

Расшифруем сообщение (19, 185, 67), пользуясь секретным ключом (25,247):

$$M_1 = (19^{25})\ mod\ 247 = 19$$
$$M_2 = (185^{25})\ mod\ 247 = 3$$
$$M_3 = (67^{25})\ mod\ 247 = 15$$

В результате расшифрования было получено исходное сообщение (19, 3, 15), то есть "СВН".

Замечания.

1. $a \equiv b(mod\ n) \Leftrightarrow a - b$ *делится на* n Числа *a* и *b* сравнимы по *mod n*, если их разность делится на *n*:
   Например,
   $6 \equiv 2\ (mod\ \ 4),\quad 15 \equiv 3(mod\ \ 6),\quad 45 \equiv 1(mod\ \ 22).$
2. Вычисления можно производить, используя правила модульной алгебры:

$a = b(mod\ n) \Rightarrow a^k = b^k(mod\ n)$
$a = b(mod\ n) \Rightarrow ac = bc(mod\ n)$

3 Вычисления нужно проводить с использование алгоритма быстрого возведения в степень:

```
степень(a, n, m):#вычисление a^n mod m
если n == 0:
      вернуть 1
   иначе
     если  n == 1:
      вернуть a
   иначе:
      p=1
      ak=a
      i=n
      пока i >0 :
         s=i%2 #остаток по модулю 2
         если  s==1:
            p=p*ak
      ak=ak*ak
      i=(i-s)/2
      вернуть p
```

Если же вычисляется остаток по модулю, то при вычислении p, ak нужно добавлять функцию нахождения остатка по модулю.

По алгоритму быстрого возведения в степень n потребуется выполнить k умножений, где $k \in [t; 2t]$, $t = \text{окрвверх}(\log_2 n)$, например, $2^6 = 64 < 121 < 128 = 2^7$, то есть t = 7 и нам потребуется от 7 до 14 умножений этого числа.

Рассмотрим на примере вычисления $(3^{121})\, mod\ 247$. Оформим поэтапное вычисление, где k – номер шага.

k	a_k	i	s	p
1	3	121	1	3
2	3·3=9	60	0	3
3	9·9=81	30	0	3
4	81·81=6561=139	15	1	3·139=170
5	139·139=55	7	1	211
6	61	3	1	27
7	16	1	1	185

В строке 4 имеем

$$a_k = 6561 = 26 \cdot 247 + 139 \equiv 139(mod\ 247)$$
$$p = 3 \cdot 139 = 417 = 1 \cdot 247 + 170 \equiv 170(mod\ 247)$$

В конце мы получили $i = 0$, поэтому последнее значение p является результатом: $3^{121} mod\ 247 = 185$.

Умножение производилось в столбцах a_k (7 умножений при возведении в квадрат) и p (5 умножений в тех строках, в которых $s = 1$), всего 12 «сложных» умножений вместо 120 = 121 – 1 при вычислении по определению.

Задание №4. Функция хеширования. Найти хеш–образ своей Фамилии, используя хеш–функцию $H_i = (H_{i-1} + M_i)^2 \bmod n$, где $n = pq$. Числа p, q взять из задания №3.

Хешируемое сообщение «САВИН». Возьмем два простых числа p=13, q=19 (см. Приложение Е). Определим $n=pq$=13*19=247. Вектор инициализации H_0 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=247, H_0=9, M_1=19, M_2=1, M_3=3, M_4=10, M_5=15.

Используя формулу

$$H_i = (H_{i-1} + M_i)^2 \bmod n,$$

получим хеш-образ сообщения «САВИН»:

$H_1=(H_0+M_1)^2$ mod n = $(23 + 19)^2$ mod 247 = 1764 mod 247 = 35
$H_2=(H_1+M_2)^2$ mod n = $(35 + 1)^2$ mod 247 = 1296 mod 247 = 61
$H_3=(H_2+M_3)^2$ mod n = $(61 + 3)^2$ mod 247 = 4096 mod 247 = 144
$H_4=(H_3+M_4)^2$ mod n = $(144 + 10)^2$ mod 247 = 23716 mod 247 = 4
$H_5=(H_4+M_5)^2$ mod n = $(4 + 15)^2$ mod 247 = 361 mod 247= 114

В итоге получаем хеш-образ сообщения «САВИН», равный 114.

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

Пусть хеш-образ Фамилии равен 114, а закрытый ключ алгоритма RSA равен (25, 247). Тогда электронная цифровая подпись сообщения, состоящего из Фамилии, вычисляется по правилу (см. Приложение Ж)

$s = 114^{25} \bmod 247 = 114$.

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

$H = 114^{121} \bmod 247 = 114$.

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.

Ответ. 1) ФГЕЛРВЕОГЖЛПЛУВРЛНСОГЗЕЛЪ
2) R1= 0010 1110 1111 1001 0011 0110 0000 1111
3) (19, 185, 67)
4) 114.
5) 114.

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.

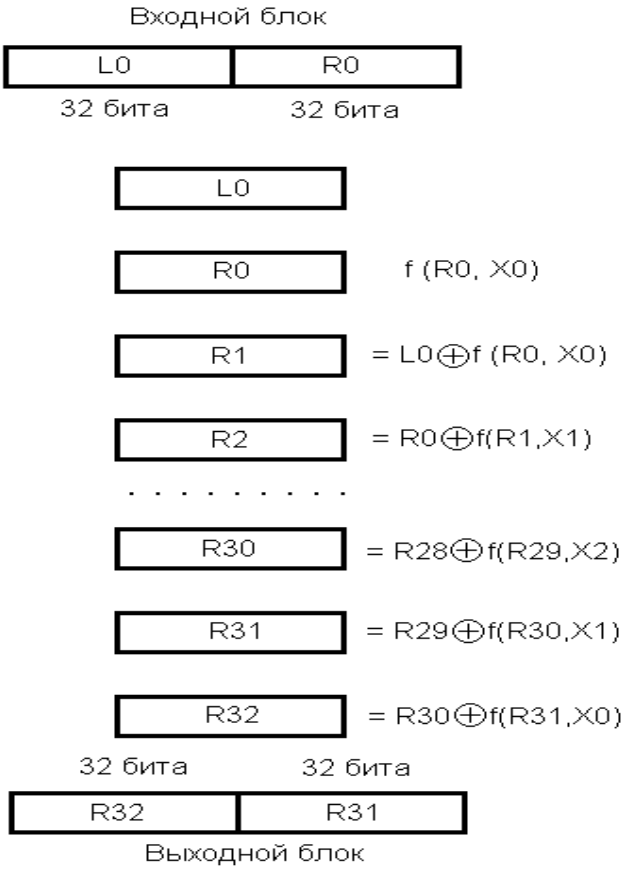
Открытые данные, подлежащие зашифрованию, разбиваются на 64-битные блоки, которые обрабатываются независимо друг от друга (Так как блоки данных шифруются независимо друг от друга, при зашифровании двух одинаковых блоков открытого текста получаются одинаковые блоки шифротекста и наоборот.). Схема обработки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

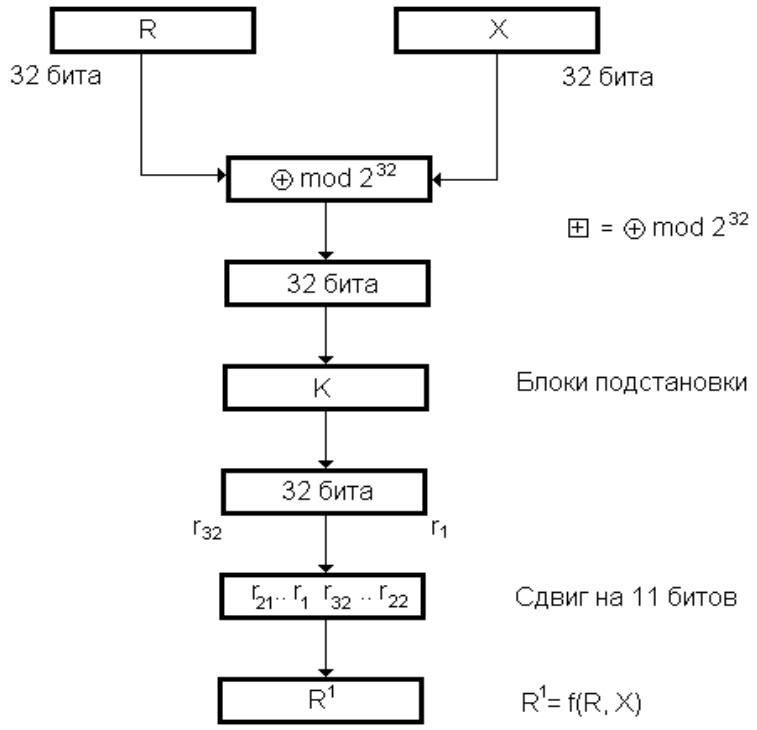


Рисунок 2 - Функция преобразования *f(R,X)* в алгоритме ГОСТ 28147-89

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	101	110	010	111	010	000	100
1	1	0	1	0	0	0	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	011	100	111	010	101	111	101
0	1	1	1	1	0	0	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 $a^x = b \pmod n$).

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 $\varphi(n)$ по формуле

$\varphi(n) = (p-1)(q-1)$.

Закрытый ключ d выбираем из условий

$d < \varphi(n)$ и

d взаимно просто с $\varphi(n)$,

т.е. d и $\varphi(n)$ не имеют общих делителей.

Открытый ключ e выбираем из условий

$e < \varphi(n)$ и

$de = 1 \pmod{\varphi(n)}$.

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

$de - 1 = \varphi(n)*k$.

В алгоритме RSA

(e, n) – открытый ключ,

(d, n) – секретный ключ.

2. Шифрование.

Исходное сообщение разбивается на блоки M_i 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

$$M^e = C \pmod n ,$$

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

$$C^d = M \pmod n .$$

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 $M' \neq M$ 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 M_i 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 X.509:

$$H_i = (H_{i-1} + M_i)^2 \bmod n ,$$

где $n = pq$, p и q – большие простые числа, H_0 - произвольное начальное заполнение, M_i - i-тый блок сообщения $M = M_1 M_2 \ldots M_k$.

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 $s = r^d \bmod n$, которое и является подписью.

Для проверки подписи получатель

1. расшифровывает подпись s на открытом ключе (e,n) отправителя, т.е. вычисляет $r' = s^e \bmod n$ 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.