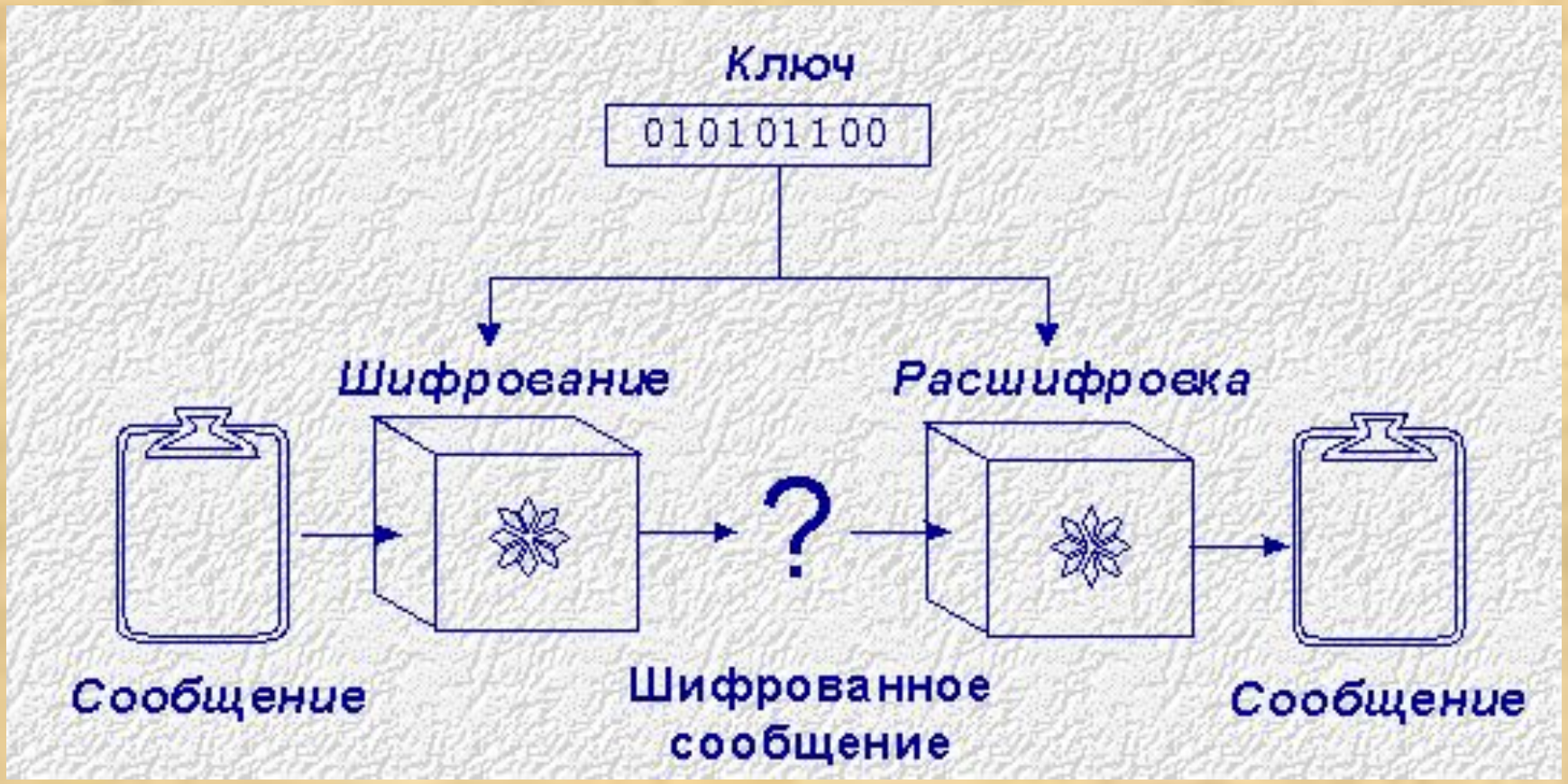
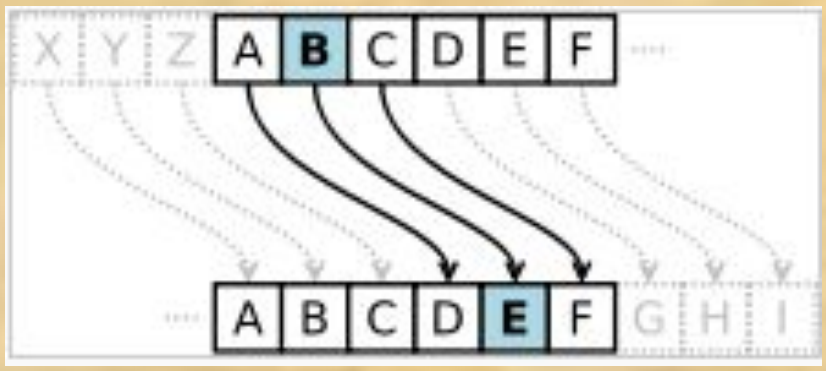
Применение простых чисел в криптографии с открытым ключом

Общий принцип криптографического сеанса:

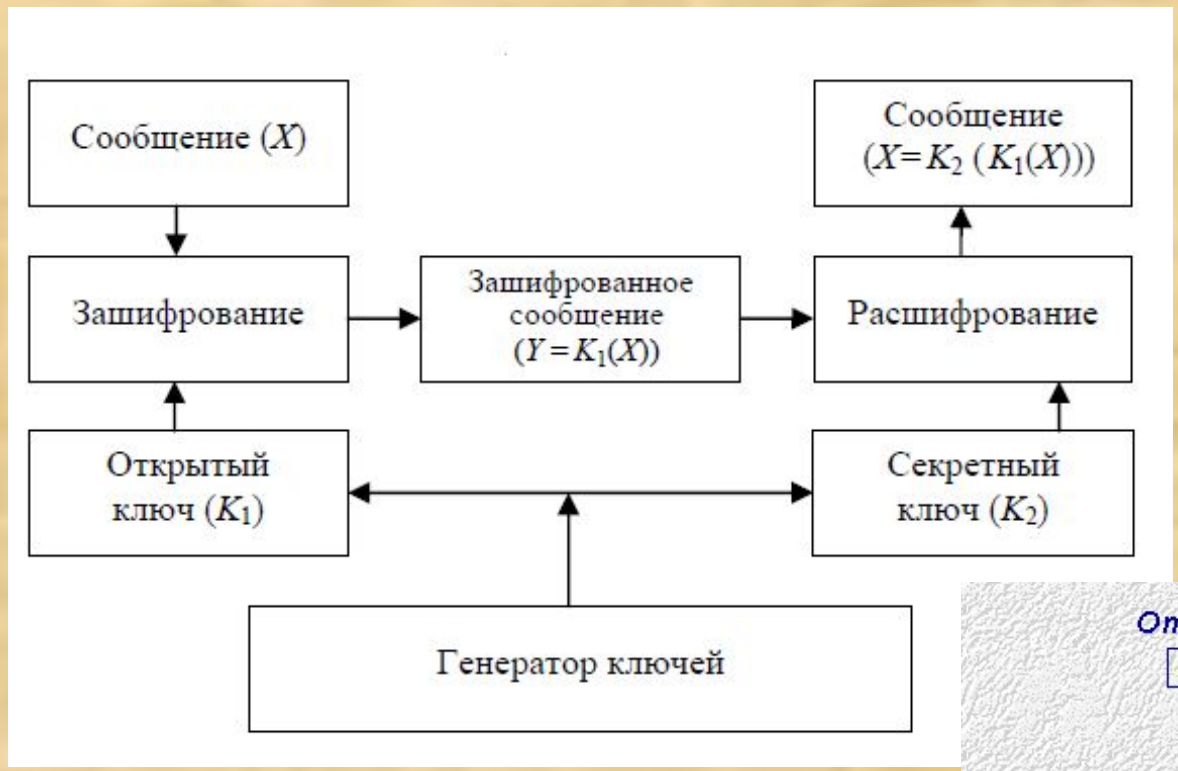


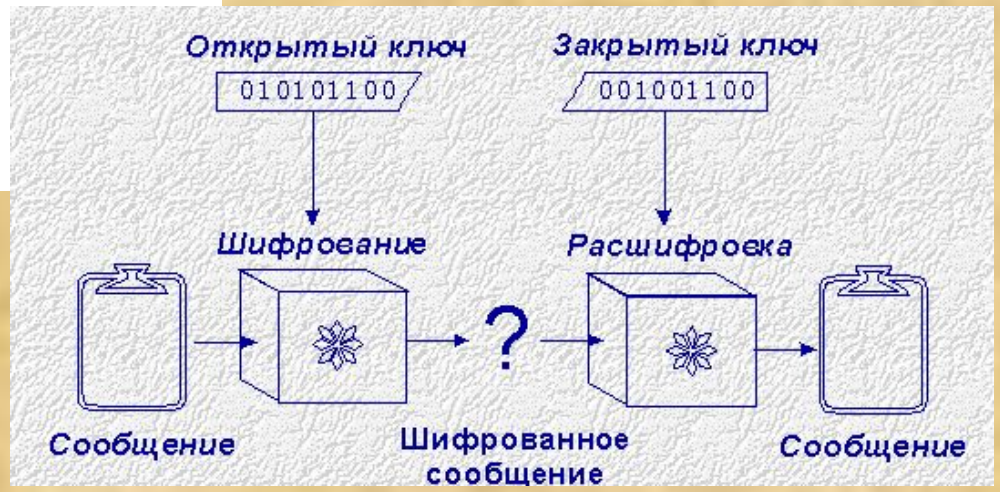
Первые криптографические алгоритмы:

	1	2	3	4	5
1	А	В	Г	Δ	Е
2	Ζ	Η	Θ	Ι	Κ
3	Λ	Μ	Ν	Ξ	Ο
4	Π	Ρ	Σ	Τ	Υ
5	Φ	Χ	Ψ	Ω	

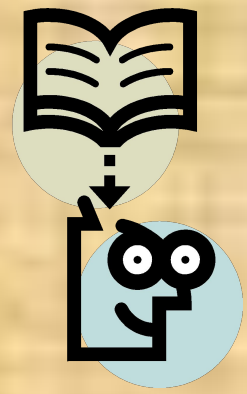


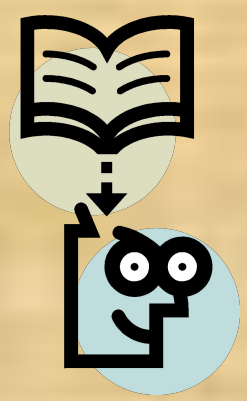
Ассиметричное шифрование с открытым ключом:





Если каждый атом во Вселенной (всего их около 2^{300}) будет являться компьютером и каждый из этих компьютеров сможет проверять 2^{300} ключей в секунду, то для того, чтобы просмотреть 1% ключей длиной 512 битов, потребуется около 2^{162} тысячелетий. В настоящее время считается, что возраст Вселенной не превышает 2^{24} тысячелетий.





Количество битов	1% общего числа ключей	50% общего числа ключей
56	1 секунда	1 минута
57	2 секунды	2 минуты
58	4 секунды	4 минуты
64	4,2 минуты	4,2 часа
72	17,9 часов	44,8 дня
80	190,9 дней	31,4 лет
90	535 лет	321 век
108	140000 тысячелетий	8 миллионов тысячелетий
128	146 миллиардов тысячелетий	8 триллионов тысячелетий

- *Определение 1.* Число *n* называется простым, если оно является натуральным и имеет ровно два различных натуральных делителя: единицу и само себя.
- *Определение 2.* Число *a* называется взаимно простым с *n*, если они не имеют никаких общих делителей, кроме ±1.

Например, 14 и 25 не являются простыми числами, но являются взаимно простыми.

Пусть произведение двух простых чисел *p* и *q* равно *n*. Положим $f=n-p-q+1$

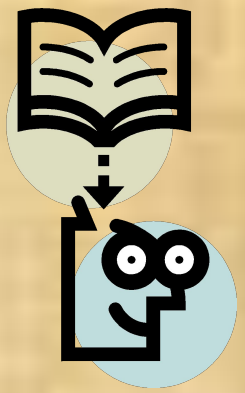
- *Лемма 1.* Для каждого числа *e,* взаимно простого с *f,* существует единственное *d,* для которого

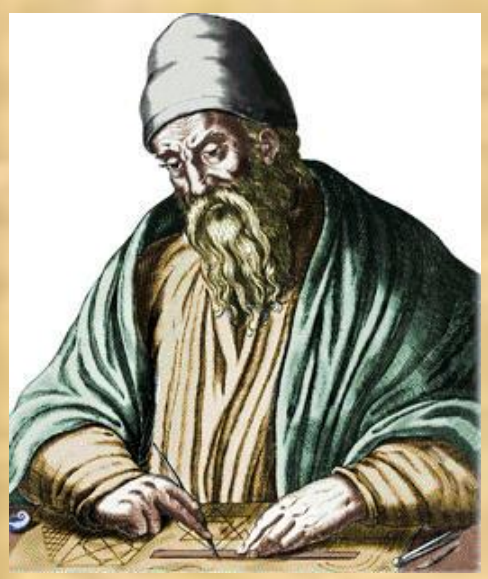
$$e \cdot d = 1 \pmod{f}.$$

- Т.е. существуют такие коэффициенты *d* и *k,* для которых

$$d \cdot e + k \cdot f = 1$$

и они единственны.

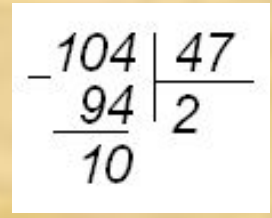




Евклид — первый математик Александрийской школы. Его главная работа «Начала» содержит изложение планиметрии, стереометрии и ряда вопросов теории чисел; в ней он подвёл итог предшествующему развитию греческой математики и создал фундамент дальнейшего развития математики.

$$f = 104,\ e = 47$$

Шаг первый:



$$\Rightarrow\ 104 = 2 \cdot 47 + 10\ \Rightarrow\ 10 = f - 2 \cdot e$$

Шаг второй: $47 = 4 \cdot 10 + 7\ \Rightarrow\ 7 = e - 4 \cdot 10\ \Rightarrow\ 7 = 9 \cdot e - 4 \cdot f$

Шаг третий: $10 = 1 \cdot 7 + 3\ \Rightarrow\ 3 = 10 - 1 \cdot 7\ \Rightarrow\ 3 = 5 \cdot f - 10 \cdot e$

Шаг четвертый: $7 = 2 \cdot 3 + 1\ \Rightarrow\ 1 = 7 - 2 \cdot 3\ \Rightarrow\ 1 = 31 \cdot e - 14 \cdot f$

$$\Rightarrow\ 31 \cdot e - 14 \cdot f = 1\ \Rightarrow\ d = 31$$

Пусть данный индивидуум ожидает получение некоторого суперсекретного сообщения от своего друга.

1) Он выбирает два простых числа – пусть это будут, например, *p = 997 и q = 1097.*

2) Он вычисляет их произведение *n = 1093709*

3) Он вычисляет *f = n-p-q+1, т.е . f = 1091616*

4) Он выбирает число *e* взаимно простое с *f.* Пусть *e = 397.*

5) Он вычисляет для него число *d* – в соответствии с алгоритмом Евклида, так, как это было сделано на предыдущем слайде. Т.е. *d* = 145777.

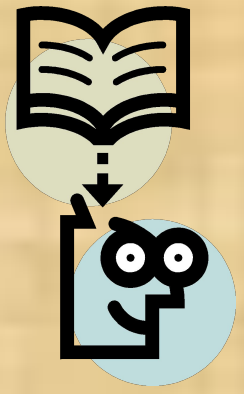
6) Он отсылает числа *n* и *e* своему другу и ждет от него ответ.

Пусть данный индивидуум – тот, кто должен послать секретное сообщение.

1) Он получает два числа: *n* = *1093709* и *e* = *397.*

2) Он кодирует отправляемый текст таким образом, чтобы он состоял из отдельных чисел в диапазоне от 1 до *n* (например, номерами букв в алфавите).

3) Каждое число текста он возводит в степень *e* по модулю *n.*



Здесь необходимо помнить, что возведение числа в большую степень сильно упрощается, если используется двоичное разложение показателя степени. Например, чтобы возвести число x в степень $29 = 11101_2$, нужно последовательно вычислить $x^{2^0}, x^{2^1}, x^{2^2}, x^{2^3}, x^{2^4}$ и перемножить нужные четыре сомножителя, не забывая о том, что числа в записи будут идти в порядке убывания степеней (предпоследний сомножитель не будет включен в общее произведение, поскольку в двоичном разложении числа 29 на предпоследней позиции стоит 0). Т.е., например $3^{29} = 3^{2^4} \cdot 3^{2^3} \cdot 3^{2^2} \cdot 3^{2^0} = 68630377364883$.

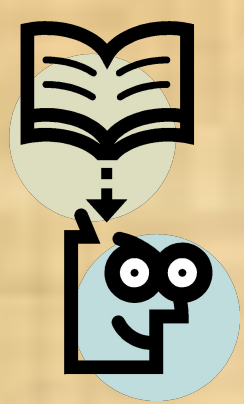
4) Полученный текст из чисел он пересылает своему другу.

Данный индивидуум получает ожидаемый текст. Это – поток чисел.

Он знает число *d* – ибо он сам его вычислил.

Каждое число полученного текста он возводит в степень *d* по модулю *n.*

Новые числа – это (в нашем учебном примере) номера нужных букв в алфавите. Переведем числа в буквы…



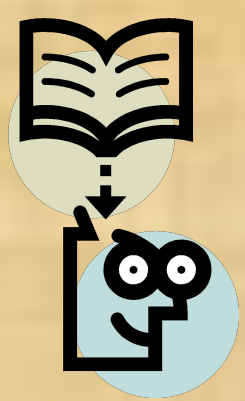
Здесь дешифрование основано на следующей лемме из теории чисел:

Лемма. Пусть a — число, взаимно простое с n. Тогда $a^f = 1 (\mathrm{mod}\, n)$.

Данный индивидуум является злоумышленником.

Он просматривает каналы связи и знает все, что касается этой переписки.

Итак, он знает числа *n* и *e* и закрытый текст. Для вскрытия текста ему понадобится число *d*. Он легко сможет найти это число, если будет знать число *f*. Для того, чтобы вычислить *f,* достаточно знать всего лишь числа *p* и *q* - два простых сомножителя числа *n*. Число *n* злоумышленнику известно. Так ли трудно найти его простые сомножители?

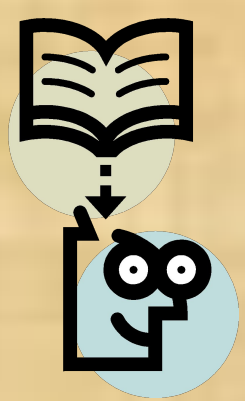


Ответ – да. Если число n достаточно велико (а в реальных алгоритмах это число действительно очень большое), то мы сталкиваемся со знаменитой проблемой простых чисел.

Первым проблему определения простых чисел поставил древнегреческий ученый Эратосфен примерно в 220 году до нашей эры, предложив один из путей определения простых чисел. С тех пор ученые постепенно продвигались вперед. Знаменитая «Гипотеза Римана» была сформулирована немецким математиком Георгом Фридрихом Бернардом Риманом в 1859 году. Согласно ей, характер распределения простых чисел может существенно отличаться от предполагаемого в настоящее время.

Рюкзак Мэркла-Хеллмана

Дана куча предметов различной массы, необходимо выяснить, можно ли положить некоторые из этих предметов в рюкзак так, чтобы масса рюкзака стала равна определенному значению?



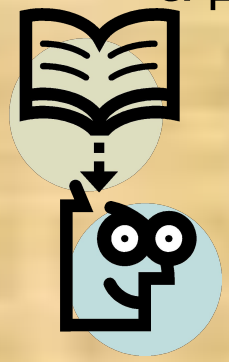
Дан набор значений M_1, M_2,..., M_n и сумма S, необходимо вычислить значения b_i, такие что

$$S = b_1M_1 + b_2M_2 + \ldots + b_nM_n$$

при этом b_i может быть либо нулем, либо единицей. Единица показывает, что предмет кладут в рюкзак, а ноль - что не кладут.

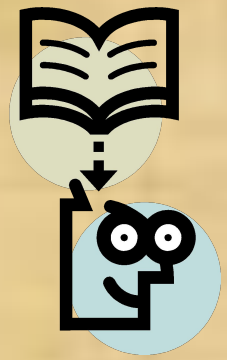
Исходный текст	111001	010110	000000	011000
Рюкзак	1 5 6 11 14 20	1 5 6 11 14 20	1 5 6 11 14 20	1 5 6 11 14 20
Шифротекст	1+5+6+20=32	5+11+14=30	0=0	5+6=11

Для рюкзака всегда используют два набора масс:
один набор является возрастающей последовательностью,
а другой – сверхвозрастающей.



Сверхвозрастающая последовательность – это последовательность, в которой каждой следующий элемент больше суммы всех предыдущих элементов.

Например, последовательность
{1,3,6,13,27,52} является сверхвозрастающей,
а {1,3,4,9,15,25} - нет.



Решение сверхвозрастающего рюкзака найти легко. Возьмем полный вес и сравним его с самым большим числом последовательности. Если полный вес меньше, чем это число, то его не кладут в рюкзак. Если полный вес больше или равен этому числу, то оно кладется в рюкзак. Уменьшим массу рюкзака на это значение и перейдем к следующему по величине числу последовательности. Будем повторять, пока процесс не закончится. Если полный вес уменьшится до нуля, то решение найдено. В противном случае решения просто нет.

Пусть данный индивидуум ожидает получение некоторого суперсекретного сообщения от своего друга.

1) Он выбирает сверхвозрастающую последовательность рюкзака, например, {2,3,6,13,27,52}

2) Он выбирает два взаимно простых числа, например, *m*=105 и *n*=31. Важно!!! *m* должно быть больше суммы всех чисел в рюкзаке!!!

3) Каждое значение сверхвозрастающей последовательности рюкзака он умножает на *n* по модулю *m*.

2*31 mod 105 = 62

3*31 mod 105 = 93

6*31 mod 105 = 81

13*31 mod 105 = 88

27*31 mod 105 = 102

52*31 mod 105 = 37

Итого: обычный рюкзак {62,93,81,88,102,37}.

4) Обычный рюкзак он пересылает своему другу.

Пусть данный индивидуум – тот, кто должен послать секретное сообщение.

1) Он получает обычный рюкзак: {62,93,81,88,102,37}.

2) Он переводит отправляемый текст в двоичную кодировку (например, с помощью метода Хаффмена) и разбивает его на блоки, равные по длине числу элементов обычного рюкзака .

Например, сверхсекретный текст
«мама мыла раму»
в бинарном виде выглядит так
0110001101011011 10
а с разбиением на блоки так:
011000 110101 101110

3) Применяет к каждому блоку ключ: рюкзак {62,93,81,88,102,37}:
011000 соответствует 93 + 81 = 174
110101 соответствует 62 + 93 + 88 + 37 = 280
101110 соответствует 62 + 81 + 88 + 102 = 333
Итак, шифротекстом будет последовательность
174 280 333

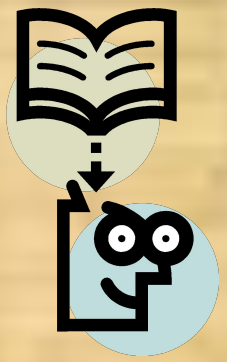
Данный индивидуум получает ожидаемый текст. Это – поток чисел.

Он находит число d – так же как и в первом случае, пользуясь алгоритмом Евклида – такое, что
$d \bullet n = 1 \pmod{m}$

В нашем примере, если $m = 105$ и $n = 31$, то $d = 61$.

Умножаем каждое число шифротекста на 61 mod 105 и применяем к полученному значению закрытый ключ {2,3,6,13,27,52}:

174*61 mod 105 = 9 = 3 + 6, что соответствует 011000
280*61 mod 105 = 70 = 2 + 3 + 13 + 52, что соответствует 110101
333*61 mod 105 = 48 = 2 + 6 + 13 + 27, что соответствует 101110



1 в современной криптографии используют так называемые гибридные криптосистемы, позволяющие ускорить события: для шифрования сообщения используется симметричный алгоритм со случайным ключом, а алгоритм с открытым ключом применяется для шифрования случайного сеансового ключа.

2 **Полезная система аутентификации пользователя, называемая *цифровой подписью.***

Спасибо
за
внимание!