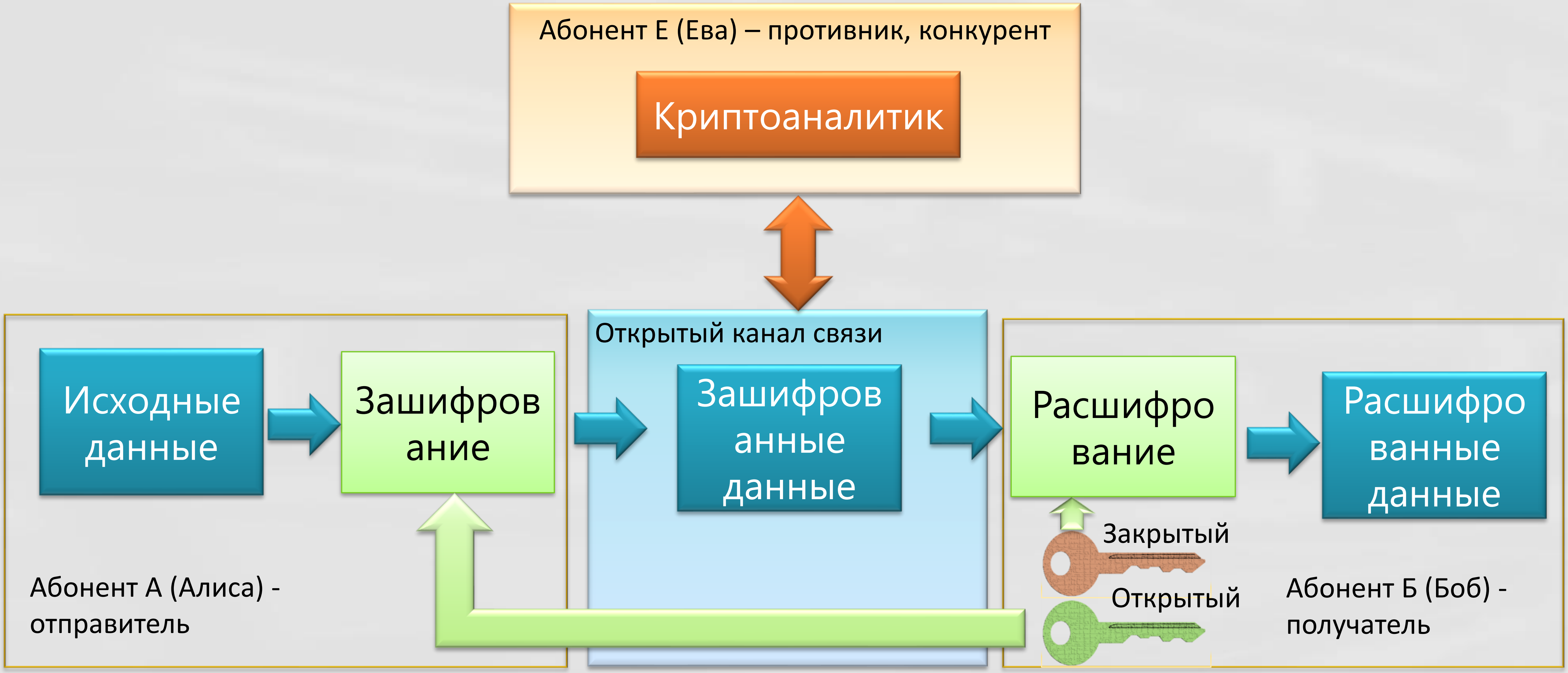
Асимметричное шифрование

Модель протокола асимметричного шифрования



Свойства асимметричного шифра

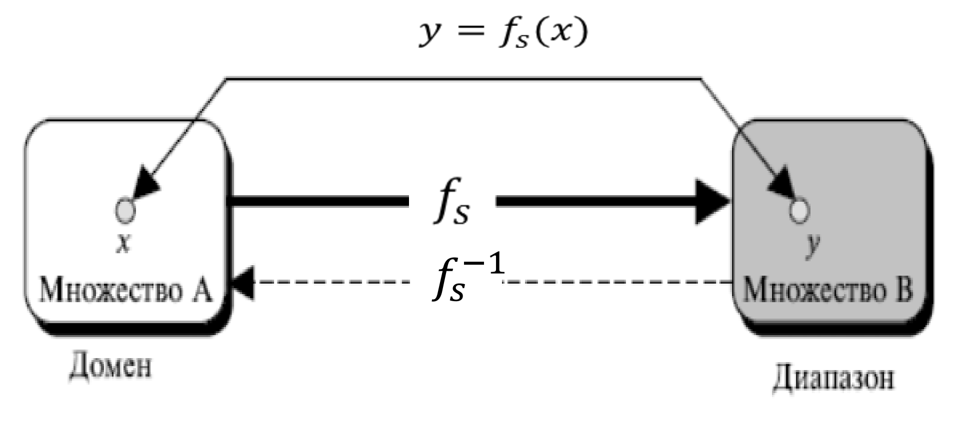
- Асимметричный шифр с помощью математических функций преобразует большие числа, представляющие скрываемые данные
- Для зашифровки и расшифровки используются различные (асимметричные) ключи, которые <u>связаны между собой математически и образуют пару</u>
- Открытый ключ (*public key*) может быть известен всем
- Закрытый ключ (*private key*) должен знать только его владелец

Требования к шифрам с открытым ключом
(Диффи и Хеллман, 1970)

- Вычислительно легко создавать пару (открытый ключ, закрытый ключ)
- Вычислительно легко, имея открытый ключ и незашифрованное сообщение, создать соответствующее зашифрованное сообщение
- Вычислительно легко расшифровать сообщение, используя закрытый ключ
- Вычислительно сложно, зная открытый ключ, определить закрытый ключ
- Вычислительно сложно, зная открытый ключ и зашифрованное сообщение, восстановить исходное сообщение

Односторонняя функция с секретом (люком)
(TOWF — Trapdoor One Way Function)

- Зная x, при любом s легко вычислить y=f_s(x)
- По известному значению y и s легко вычислить x=$f_s^{-1}(y)$
- Сложно вычислить x=$f_s^{-1}(y)$ по известному y, если секрет s не известен



Пример TOWF : Задача о рюкзаке

- Зная подмножество предметов, уложенных в рюкзак, легко подсчитать его суммарный вес, но, зная только вес рюкзака, и веса отдельных предметов сложно определить подмножество предметов в рюкзаке:
  - $s(X,B) = \sum_1^n b_i \times x_i$, n – количество предметов
  - $X = (x_1, x_2, \dots x_n)$ – бинарное представление подмножества предметов (1- уложен, 0-оставлен)
  - $\mathrm{B} = (b_1, b_2, \dots b_n)$ – набор весов предметов
- Для произвольного набора чисел b_i задача восстановления X по s является вычислительно сложной
- $s(X,B)$ можно рассматривать, как шифровку $X = (x_1, x_2, \dots x_n)$

Секрет предметов из рюкзака

- В частном случае задача имеет полиномиальный алгоритм решения, когда последовательность $(b_1, b_2, \dots\ b_n)$ является супервозрастающей (*superincreasing*): $b_k \geq \sum_1^{k-1} b_i$
- Алгоритм состоит в просмотре списка предметов в <u>порядке убывания</u> их весов и принятия решения для каждого предмета относительно возможности укладки в рюкзак
- Возьмем за основу этот алгоритм, но только добавим «секрет»:
  - Выбираем число $q > \sum_1^n b_i$ и число r, взаимно простое с q
  - Вычисляем $a_i = (r * b_i) \bmod q$ – это открытый ключ
  - Держим в секрете r, q и $(b_1, b_2, \dots\ b_n)$ – это закрытый ключ

Ранцевое шифрование Меркеля — Хеллмана

- Зная подмножество предметов, уложенных в рюкзак, легко подсчитать его суммарный вес (*зашифрование*), но, зная только вес рюкзака (*шифровку*), и веса отдельных предметов (*открытый ключ*), сложно определить подмножество предметов в рюкзаке, если секрет (*закрытый ключ*) неизвестен:
  - $s(X, A) = \sum_1^n a_i \times x_i$, n – количество предметов
  - $X = (x_1, x_2, \dots\ x_n)$ – бинарное представление подмножества предметов (1- уложен, 0-оставлен)
  - $A = (a_1, a_2, \dots\ a_n)$ – специальный набор весов предметов
- Только обладатель секрета r, q и $(b_1, b_2, \dots\ b_n)$: $b_k \geq \sum_1^{k-1} b_i$ сможет применить полиномиальный алгоритм для определения подмножества предметов в рюкзаке

Решение задачи при знании секрета

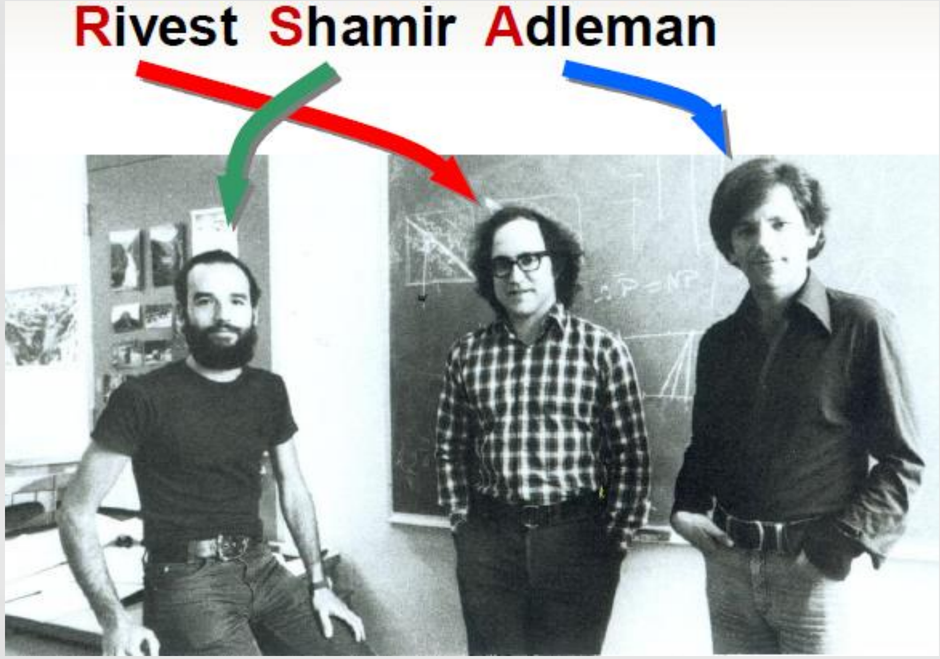
- Имеем значение функции $s = \sum_1^n a_i \times x_i$, знаем $(a_1, a_2, \dots\ a_n)$ и нужно определить $(x_1, x_2, \dots\ x_n)$
- Находим r'такое, что $r' \times r \equiv 1\ mod\ q$ – мультипликативная инверсия r (используем расширенный алгоритм Евклида для решения уравнения $r' \times r + t \times q = 1$)
- Вычисляем $s' = (r' \times s)\ mod\ q$
- Можно показать, что $s' = (\sum_1^n b_i \times x_i\ ) mod\ q = \sum_1^n b_i \times x_i$
- Окончательно решаем частный случай задачи о рюкзаке с супервозрастающей последовательностью весов предметов и определяем $(x_1, x_2, \dots\ x_n)$

Методы асимметричного шифрования

Шифр RSA

Историческая справка

- RSA (Rivest, Shamir, Adleman) – создатели шифра Рональд Райвест, Ади Шамир и Леонард Адлеман) из Массачусетского Технологического Института
- Шифр разработан в 1977 году и основан на проблеме разложения больших целых чисел на простые множители
- В 1982 году Ривест, Шамир и Адлеман организовали компанию RSA Data Security
- В 1990 году алгоритм начинает использовать министерство обороны США



Шифр RSA

- Шифр RSA базируется на следующих двух фактах из теории чисел:
  - задача проверки числа на простоту является сравнительно легкой;
  - задача разложения чисел вида $n = p*q$ (p и q — простые числа) на множители является очень трудной, если мы знаем только n, а p и q — большие числа (это так называемая задача факторизации)
- Шифр RSA представляет собой блочный алгоритм шифрования, где зашифрованные и незашифрованные данные должны быть представлены в виде целых чисел между 0 и $n - 1$

RSA генерация ключей

- Выбираются два больших простых числа *p* и *q*
- Вычисляется *n=p*q*
- Выбирается произвольное число *e (e<n)*, взаимно простое с $(p-1)\times(q-1)$
- Вычисляется d, такое, что $e \times d \equiv 1\, mod (p-1)\times(q-1)$ решением в целых числах уравнения (расширенный алгоритм Евклида) относительно *d* и *y* :

$$e \times \boldsymbol{d} + (p-1)\times(q-1)\times \boldsymbol{y} = \text{НОД}\big(e,(p-1)*(q-1)\big) = 1$$

- Пара чисел *(e, n)* объявляются открытым ключом,
- Закрытым ключом выбирается *d* , <u>*p* и *q* нужно уничтожить</u>

RSA зашифрование

- Открытый текст разбивается на блоки m_i размером $k \leq [log_2 n]$ бит. Блоки интерпретируются, как числа из диапазона *(0; 2^k -1)*
- Ключ шифрации (открытый ключ) – пара чисел *(e, n)*
- Каждый блок открытого текста преобразуется в шифротекст по формуле:

$$c_i = (m_i{}^e) mod\, n$$

RSA расшифрование

- Ключ для расшифровки сообщения – *d* (закрытый ключ)
- Блок шифротекста преобразуется в открытый текст по формуле:

$$m_i = ({c_i}^d) mod\ n$$

- Доказательство основано на теореме Эйлера: если *n* представимо в виде произведения простых чисел *p* и *q*, то для *x* (взаимно простого *c n*) справедливо:

$$(x^{(p-1)\times(q-1)}) mod\ n=1$$

RSA доказательство корректности расшифрования

- Возведем в $(-y)$ обе части уравнения $(x^{(p-1)\times(q-1)}) mod\ n = 1$
- В полученном равенстве

$$(x^{(-y)\times(p-1)\times(q-1)}) mod\ n = 1^{(-y)}$$

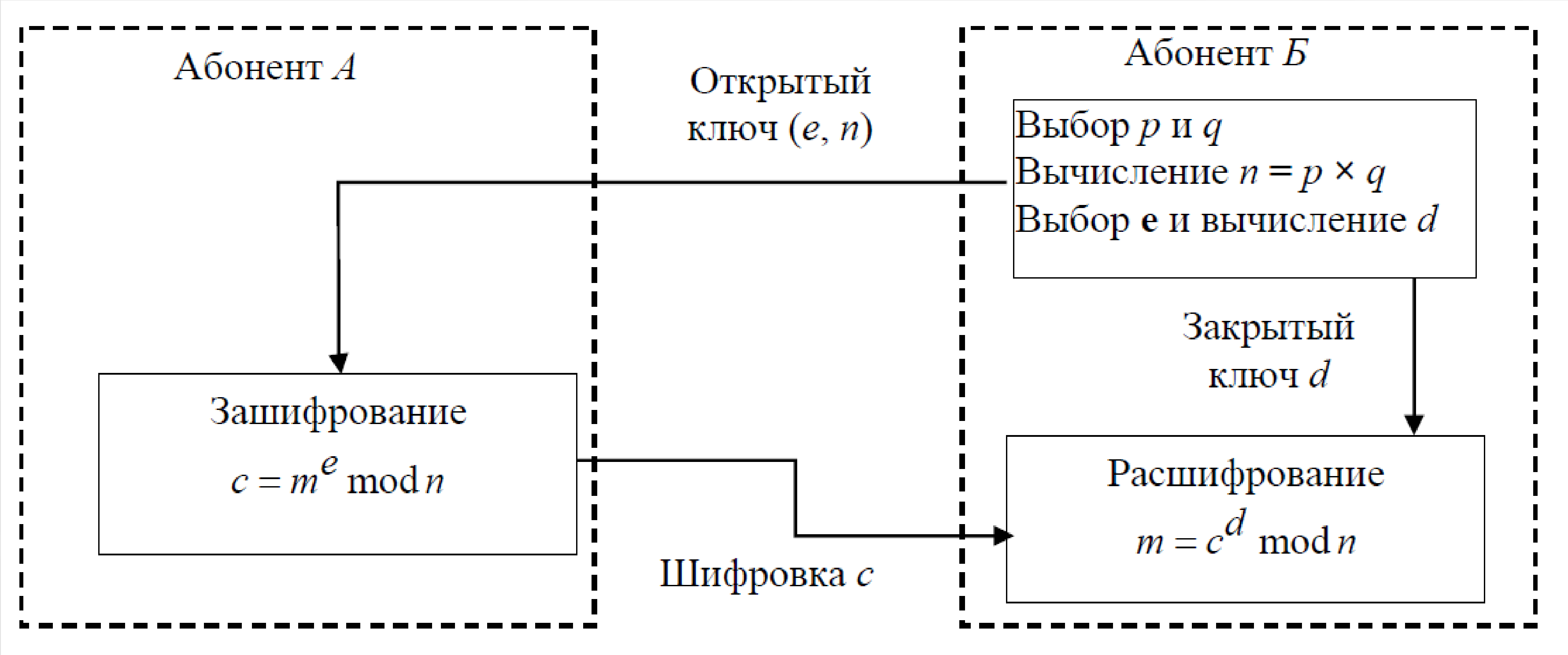
умножим на x левую и правые части. В итоге получаем:

$$(x^{1-y\times(p-1)\times(q-1)}) mod\ n = x \times 1^{(-y)}$$

- Поскольку $1-(p-1)\times(q-1)\times y = e \times d,$ то при замене x на m_i получаем:

$$((m_i)^{e\times d}) mod\ n = ((m_i{}^{e})^{d}) mod\ n = ((c_i)^{d}) mod\ n = m_i$$

Протокол шифрования на основе RSA



- Секретный и открытый ключи RSA равноправны - каждый из ключей (d или e) может использоваться как для зашифрования, так и для расшифрования
- Совпадающие блоки зашифровываются одинаково (как в режиме электронной кодовой книги)

Обучающий ролик по протоколу RSA

- https://www.youtube.com/watch?v=vooHjWxmcIE

Алгоритм быстрого возведения в степень

- Вычисляет функцию $y = a^x mod\, n$
- Представим x $= m_k 2^k$+$m_{k-1}2^{k-1}$+…+$m_1 2$+m_0, где m_k=1,$m_i \in \{0,1\}$
- Тогда a^x=$a^{\left(\left(\ldots\left(\left(m_k * 2+m_{k-1}\right)*2+m_{k-2}\right)*2+\cdots\right)*2+m_1\right)*2+m_0}$=
- $((\ldots(((a^{m_k})^2 * a^{m_{k-1}})^2 \ldots)^2 * a^{m_1})^2 a^{m_0}$
- Получаем мультипликативный аналог схемы Горнера:
- $$\begin{cases} s_1 = a\, mod\, n \\ s_{i+1} = s_i{}^2 \quad * a^{m_{k-i}} \text{mod n} \\ i = 1, \ldots, k \end{cases}$$
- Сложность алгоритма $o(log_2 x)$

Безопасность RSA

- Базируется на предположении, что модуль *n* настолько большой, что разложение на множители в разумное время неосуществимо
- Авторы RSA рекомендовали использовать следующие размеры модуля *n*: 768 бит - для частных лиц; 1024 бит - для коммерческой информации; 2048 бит - для особо секретной информации
- В настоящее время эти значения следует удвоить

Атака разложения на множители

- **Цель – вычисление закрытого ключа получателя**
- Если для заданного n найдены большие простые числа p и q, такие, что $p*q = n$, то можно вычислить

$$(p-1) \times (q-1)$$

- Решается в целых числах уравнение (расширенный алгоритм Евклида) относительно *d* и *y* :

$$e \times \boldsymbol{d} + (p-1) \times (q-1) \times \boldsymbol{y} = 1$$

- Находим закрытый ключ d

Метод Ферма разложение на множители

- Основан на факте, что если найдены x и y такие, что n=x^2-y^2, то найдено и разложение n=a*b, где $a = (x + y), b = (x - y)$
- Ищем y^2 =x^2-n, изменяя значение x:

```
Разложение_ на_ множители Ферма (n)    // n - раскладываемое
число
{
  x ← ⌈√n⌉ // наименьшее целое, большее, чем √n
  while (x < n)// наименьшее целое, большее, чем
    {
      w ← x²–n
      if (w полный квадрат числа) y ← √w; a ←
x + y; b ← x − y; return a and b
      x ← x + 1
    }
}
```

Атака общего модуля

- **Цель: вычисление закрытого ключа абонента**
- Сообщество абонентов использует единый модуль *n=p*q*
- Администратор предоставляет каждому абоненту открытый ключ (e_i, n) и закрытый ключ d_i
- Если нарушитель Е принадлежит сообществу, то знание e_E , d_E позволяет ему за полиномиальное время $O(log_2 n^3)$ получить разложение *n=p*q* (это доказано)
- Тогда перехватив шифровку C_A и, зная открытый ключ (e_A, n) , можно вычислить секретный ключ d_A абонента А
- **Противодействие - каждый абонент должен использовать свой собственный модуль**

Атака с выборкой зашифрованного текста

- **Цель- получение открытого текста сообщения**
- Нарушитель E перехватывает шифровку $C = P^e mod\ n$ для получателя B
- Нарушитель имеет возможность обманом («ослепление») получить от B расшифровку (подпись) специально созданного текста

  $Y = C \times X^e mod\ n$ и получает $Z = Y^d mod\ n$
- Нарушитель составляет уравнение:

$$Z = Y^d mod\ n = (C \times X^e)^d mod\ n = \left(C^d \times X^{ed}\right) mod\ n = (P \times X) mod\ n$$

- В итоге имеем $P = Z \times X^{-1} mod\ n$ и с помощью расширенного алгоритма Евклида находится мультипликативная инверсия X^{-1} и исходное сообщение P

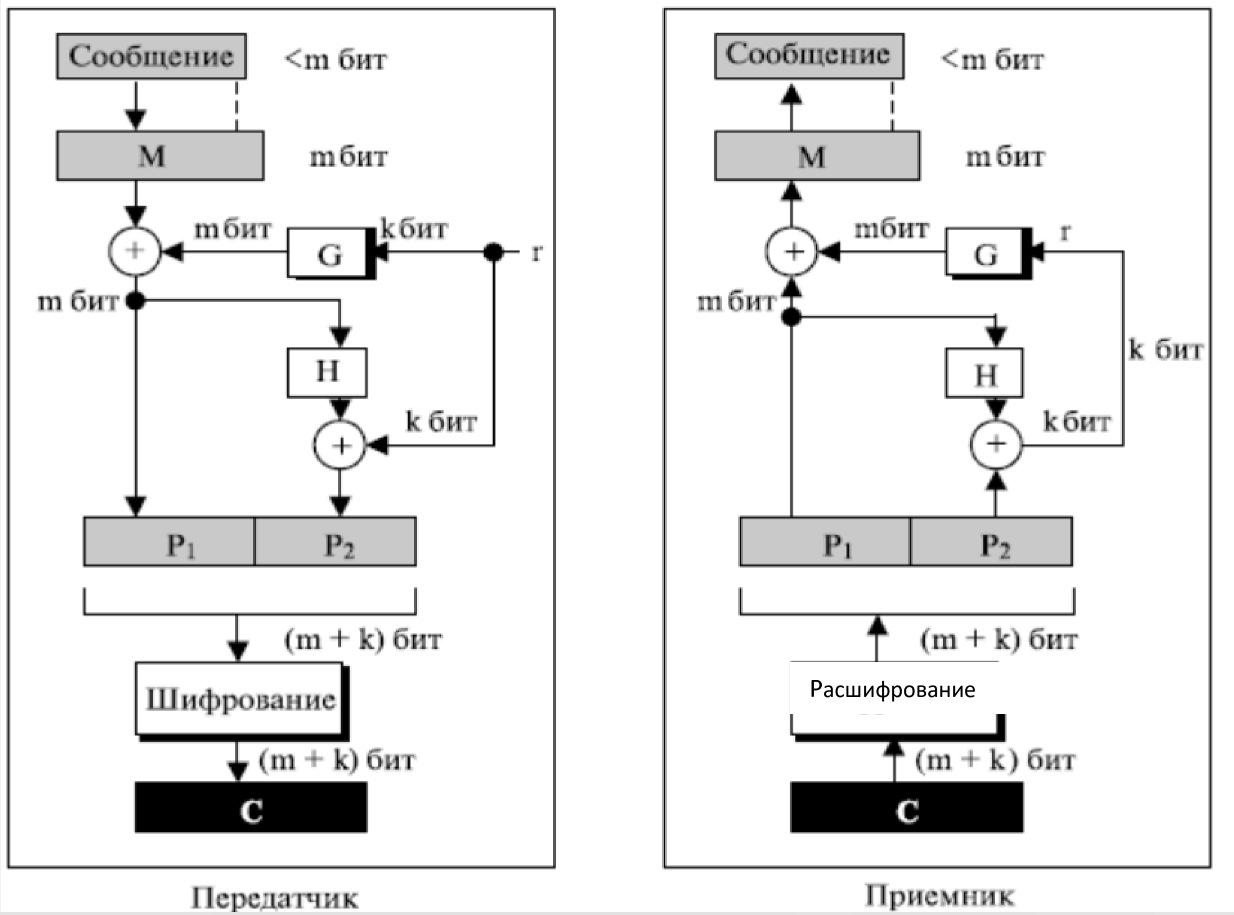
Атака при малом показатели степени (ключе) шифрования

- **Атака широковещательной передачи с целью получения открытого текста сообщения**
  - Отправитель передает одно и то же сообщение группе получателей с тем же самым ключом шифрования *e*
  - Пусть *e=3* и используются модули n_1 , n_2 , n_3 . Тогда C_1=$P^3 mod n_1$, C_2=$P^3 mod n_2$, C_3=$P^3 mod n_3$
  - Применяя китайскую теорему об остатках к этим трем уравнениям, можно найти $C' = P^3 mod\, n_1 n_2 n_3$
  - Так как $P^3 < n_1 n_2 n_3$, $C' = P^3$, то и *P* можно найти с помощью обычной арифметики
- **Противодействие: Генерировать сообщения** $f_i(P) = (i * 2^i + P)$

Атаки исходного текста

- Явная атака сообщения
  - Явное сообщение — сообщение, которое зашифровано само в себя (не может быть скрыто). Доказано, что явные сообщения есть при любом ключе.
  - **Программа шифровки должна всегда проверять, является ли вычисленный зашифрованный текст таким же, как исходный текст**
- Атака короткого сообщения
  - В этом случае нарушитель может зашифровать все возможные исходные сообщения, пока результат не будет совпадать с перехваченным зашифрованным текстом
  - **Рекомендуется дополнять исходный текст случайными битами прежде начала шифрования (метод OAEP см. далее)**

Оптимальное асимметричное дополнение шифрования (OAEP — Optimal Asymmetric Encryption Padding)



- Используем двухъячеечную сеть Фейстеля
- Сообщение дополняется нулями до m бит
- Генерируется случайное k-битное число r
- Вычисляется маска G(r), где G() односторонняя функция, и маскированный текст P_1
- Вычисляется дополнение P_2, с использованием односторонней функции H()
- Обратимость схемы основано на свойстве XOR

Рекомендации по выбору параметров RSA

- Число битов для n должно быть, по крайней мере, 1024. Это означает, что n должно быть приблизительно 2^{1024} или 309 десятичных цифр
- Два простых числа p и q должны каждый быть по крайней мере 512 битов. Это означает, что p и q должны быть приблизительно 2^{512} или 154 десятичными цифрами
- Значения p и q не должен быть очень близки друг к другу
- p – 1 и q – 1 должны иметь по крайней мере один большой простой сомножитель
- Модуль n не должен использоваться совместно.
- Значение e должно быть $2^{16} + 1$ или простым числом, близким к этому значению
- Если произошла утечка закрытого ключа d, нужно немедленно изменить n, так же e и d.
- Короткие сообщения должны быть дополнены процедурой OAEP

Практическое использование RSA

- Открытое шифрование на базе алгоритма RSA применяется в популярном пакете шифрования PGP, операционной системе Windows, различных Интернет-браузерах, банковских компьютерных системах.
- RSA является полезным для коротких сообщений. В частности различные международные стандарты шифрования с открытым ключом и формирования цифровой подписи используют RSA в качестве основного алгоритма (S/MIME, TLS/SSL, IPSEC/IKE и др.)

Методы асимметричного шифрования

Шифр Рабина

Историческая справка

- Данный алгоритм был опубликован в январе 1979 Майклом О. Рабином (*Michael Oser Rabin*) израильский математик, лауреат премии Тьюринга и многих других премий.
- Шифр основан на проблеме извлечения квадратного корня по модулю составного числа : $x^2 \equiv \mathrm{a} \bmod \mathrm{n}$
- Впервые было доказано, что эта проблема столь же трудна, что и факторизация больших целых чисел

Шифр Рабина (Rabin)

- Шифр является вариантом криптосистемы RSA, только базируется на квадратичных сравнениях
- Шифр можно условно представить, как криптографическую систему RSA, в которой значениям e и d присвоены значения *e = 2* и *d = ½:*

$$c_i = ((m_i)^e) mod\, n \qquad m_i = ((c_i)^d) mod\, n$$

- Шифр представляет собой блочный алгоритм шифрования, где зашифрованные и незашифрованные данные должны быть представлены в виде целых чисел между 0 и n -1 для некоторого n

Rabin генерация ключей

- Выбираются два случайных числа *p* и *q* с учётом следующих требований:
  - числа должны быть простыми и большими с примерно одинаковой разрядностью
  - простые числа *p* и *q* могут быть представлены либо в форме $4k + 3$, либо в форме $4k + 1$ Рекомендуется применять форму $4k + 3$, для того чтобы сделать расшифровку намного проще
- Вычисляется число $n = p * q$
- Число *n* объявляется открытым ключом
- Пара чисел (p , q) образуют закрытый ключ

Rabin зашифрование

- Открытый текст разбивается на блоки m_i размером $k \leq [log_2 n]$ бит. Блоки интерпретируются, как числа из диапазона $(0; 2^k -1)$
- Ключ шифрации (открытый ключ) – число n
- Каждый блок открытого текста преобразуется в шифротекст по формуле:

$$c_i = (m_i)^2 mod\, n$$

- Примечание: *Шифрование очень простое и нуждается только в одном умножении, что может быть сделано быстро. Это выгодно, когда ресурсы ограничены: например, при использовании карт с интегральной схемой, содержащей микропроцессор с ограниченной памятью*

Rabin расшифрование

- Ключ для расшифровки сообщения – числа p и q (закрытый ключ)
- Блок шифротекста преобразуется в открытый текст решением задачи нахождения квадратичного вычета:

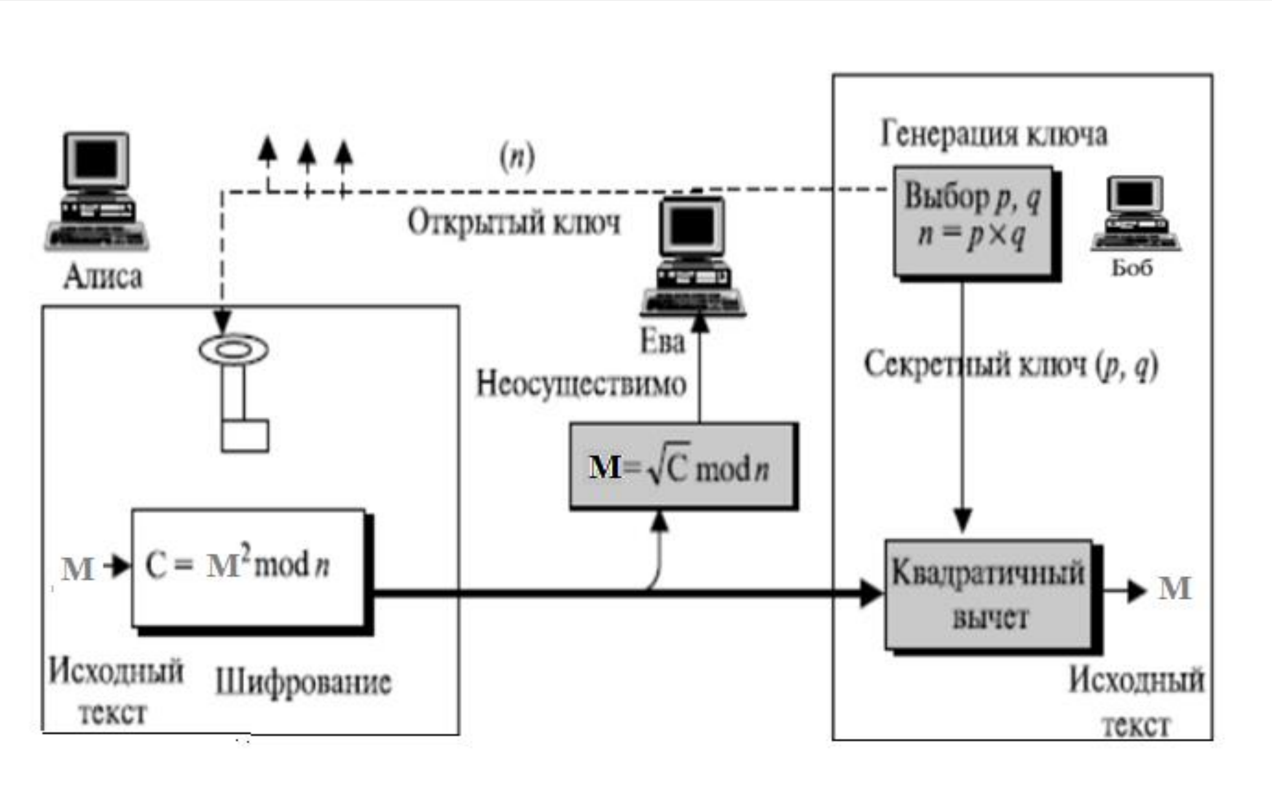
$$m_i^2 \equiv c_i \; mod \; (p*q)$$

- Решение базируется на применении расширенного алгоритма Эвклида и китайской теореме об остатках

Rabin расшифрование

- С учетом ограничения $p \equiv q \equiv 3mod4$ вычисляются:
  - $r_{1,2} = \pm c_i^{(p+1)/4}\ mod\ p$ – решения уравнения $r^2 \equiv c_i\ mod\ p$
  - $s_{1,2} = \pm c_i^{(q+1)/4}\ mod\ q$ – решения уравнения $s^2 \equiv c_i\ mod\ q$
- Решаются 4 системы уравнений (китайская теорема об остатках):
  - $m_i^2 \equiv r_1\ mod\ p \qquad m_i^2 \equiv s_1 mod\ q$
  - $m_i^2 \equiv r_1\ mod\ p \qquad m_i^2 \equiv s_2 mod\ q$
  - $m_i^2 \equiv r_2\ mod\ p \qquad m_i^2 \equiv s_1 mod\ q$
  - $m_i^2 \equiv r_2\ mod\ p \qquad m_i^2 \equiv s_2 mod\ q$
- Решением являются 4 значения m_i и только одно решение правильное

Протокол шифрования на основе Rabin-шифра



- Криптосистема Рабина не детерминирована — дешифрование создает четыре одинаково вероятных исходных сообщения
- Сложность расшифрования в системе Рабина такая же, как и у процедуры разложения больших чисел *n* на два простых сомножителя *p и q* (система также безопасна, как и RSA)

Методы асимметричного шифрования

Шифр Эль-Гамаля

Историческая справка

- Схема была предложена в 1985 году Тахером Эль-Гамалем (Taher Elgamal), египетским криптографом
- В отличие от RSA алгоритм Эль-Гамаля не был запатентован и, поэтому, стал более дешевой альтернативой
- Схема Эль-Гамаля лежала в основе бывших стандартов электронной цифровой подписи в США (DSA) и России (ГОСТ Р 34.10-94).

Шифр Elgamal

- Шифр является усовершенствованием системы Диффи-Хеллмана
- Шифр основан на вычислении дискретных логарифмов в конечном поле :
  - $\text{Пусть } y = g^x mod\ p$
  - Вычислительно трудно найти *x* при известных *y, g, p*
- Проблема вычисления дискретного логарифма имеет такую же сложность, как проблема разложения на множители

Elgamal генерация ключей

- Генерируется случайное простое число *p*
- Выбирается целое число *g* такое, что *1< g< p,* и *g*-порождающий элемент циклической группы (генератор) порядка *p,* для которого справедливо: $g \bmod p, g^2 \bmod p, g^3 \bmod p \ldots g^{p-1} \bmod p$ являются различными целыми из *[1,p-1]*
- Выбирается случайное целое число *x* такое, что *1 < x < p*
- Вычисляется $y = g^x \bmod p$
- Открытым ключом объявляется тройка *(p, g, y)*
- Закрытым ключом назначается число *x*

Elgamal зашифрование

- Открытый текст разбивается на блоки m_i размером $\mathrm{k} = [\log_2 \mathrm{p}]$ бит. Блоки интерпретируются, как числа из диапазона *(0;* 2^k *-1)*
- Ключ шифрации (открытый ключ) – тройка *(p, g, y)*
- Выбирается сессионный ключ-случайное целое число *k, 1<k<p-1*
- Каждый блок открытого текста преобразуется в пару чисел *(a,b)*:

$$a = \left(g^k\right) mod\ p \qquad b = (y^k * m_i\ )mod\ p$$

- Эта пара чисел *(a,b)* является блоком шифротекста c_i
- Длина шифротекста вдвое больше длины исходного сообщения

Elgamal расшифрование

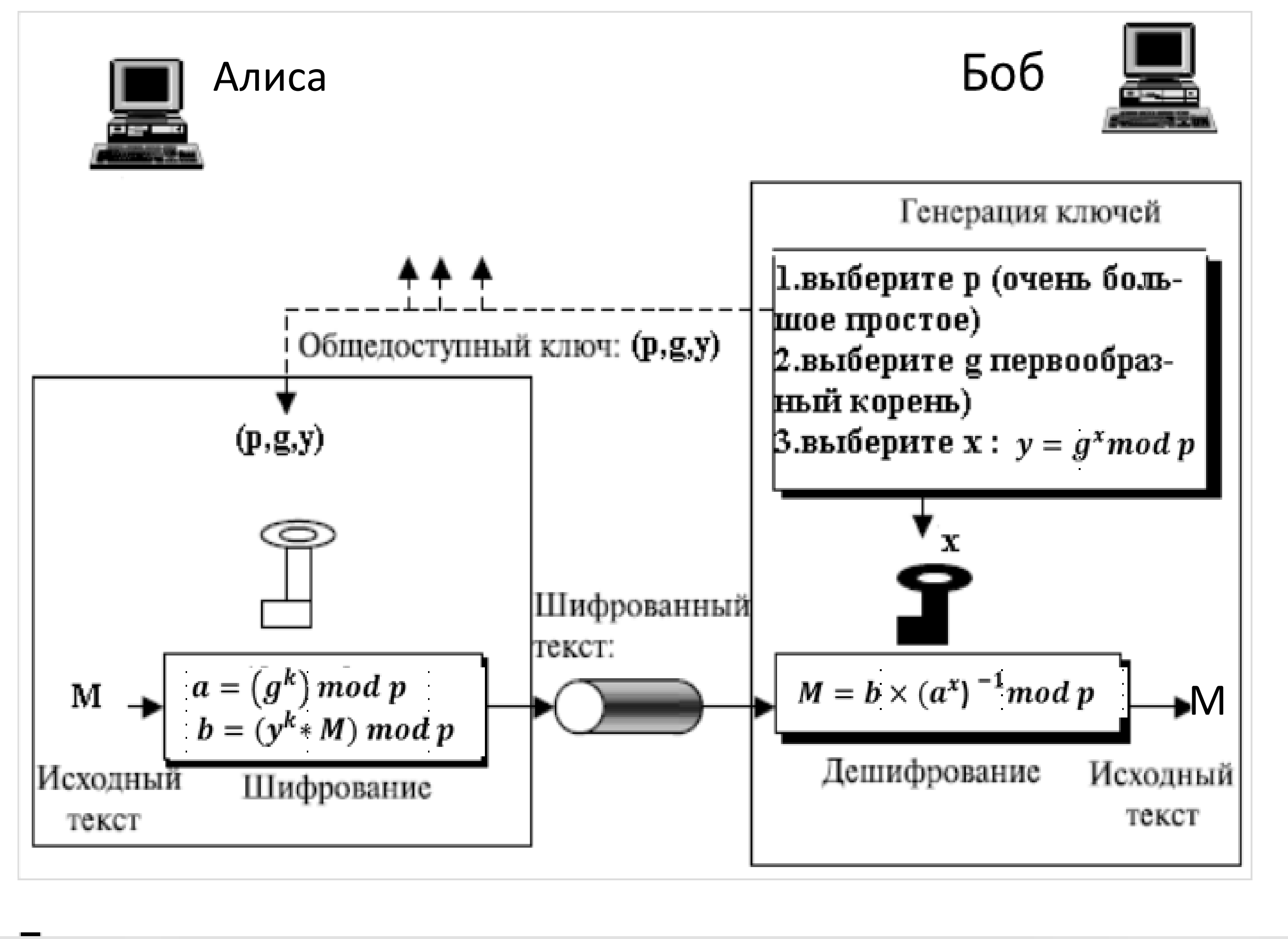
- Ключ расшифрования – число x (закрытый ключ)
- Блок шифротекста преобразуется в открытый текст по формуле:
  - $m_i = b \times (a^x)^{-1} mod\, p$
  - Поскольку $(a^x)^{-1} \equiv g^{-kx} mod\, p$ (подстановка ранее определенного a), имеем (подстановка ранее определенного b):

    $b \times (a^x)^{-1} \equiv (y^k * m_i) \times g^{-kx} \equiv (g^{kx} * m_i) \times g^{-kx} \equiv m_i\, mod\, p$
- Для практических вычислений используется выражение:

  $m_i = b \times (a^x)^{-1} mod\, p = b \times a^{(p-1-x)} mod\, p$ (т. к. $a^{(p-1)} \equiv 1 mod\, p$

согласно малой теоремы Ферма)

Протокол шифрования на основе шифра EG- шифра



- Отправитель создает маску $y^k = g^{xk}$, которая скрывает значение открытого текста M.
- Получатель создает точную копию маски $a^x = g^{kx}$ и инвертирует ее (мультипликативная инверсия), чтобы снять маску с шифротекста
- Отправителю остается неизвестным число *x*, а получателю остается неизвестным число *к*

Пример

- Ключ: $p=11$, $g=2 (2^{10} \equiv 1 \bmod 11)$, $x = 3$, $y = g^x \bmod p = 2^3 \bmod 11 = 8$.
  - Открытый ключ $(p,g,x)=(11,2,8)$
  - Закрытый ключ $x=3$
- Зашифрование открытого текста $\boldsymbol{m_i=7}$
  - $k=4$, $a = (g^k) \bmod p = 2^4 \bmod 11 = 5$; $b = (y^k * m_i) \bmod p = (4096 \times 7) \bmod 11 = 6$
  - Зашифрованный текст $(a,b)=(5,6)$
- Расшифрование:
  - $b \times a^{(p-1-x)} \bmod p = 6 \times 5^7 \bmod 11 = 6 \times 3 \bmod 11 = \boldsymbol{7 = m_i}$

Безопасность шифра

- Чтобы шифр Эль-Гамаля был безопасен, модуль p должен содержать по крайней мере 300 десятичных цифр
- Модуль p или случайное число k, которое отправитель использует для зашифровки, должны обновляться для каждой передачи сообщения, чтобы предотвратить атаку знания исходного текста:
  - $b = (y^k * M) mod\, p \quad b' = (y^k * M') mod\, p$ и пусть M стало известно
  - Тогда $y^k = b \times M^{-1} mod\, p$ и $M' = b' \times (y^k)^{-1} mod\, p$
- Шифр Эль-Гамаля может использоваться всякий раз, когда может использоваться RSA, т.е. шифрования и дешифрования маленьких сообщений

Атака на дискретный логарифм

- Предложена советским математиком Александром Осиповичем Гельфондом в ещё в 1962
- Использован представление степени $x = x_1 S + x_2,$

  где $x_1 \le S-1,\ x_2 \le S-1.$ целые неотрицательные числа,

  а $S = \lceil \sqrt{p-1} \rceil$ (ближайшее большее целое)
- Вычисляются следующие S чисел:

λ	0	1	2	$\dots$	$S-1$	$-S$
$g^\lambda \mod p$	g^0	g^1	g^2	$\dots$	g^{S-1}	g^{-S}

Атака на дискретный логарифм (продолжение)

- Предположим, что x_1=λ, λ=0,…,S-1
- Тогда $x = \lambda S + x_2$ и $y = g^{\lambda S + x_2}$mod p
- Если число $yg^{-\lambda S}$mod p = g^{x_2}mod p содержится в таблице,

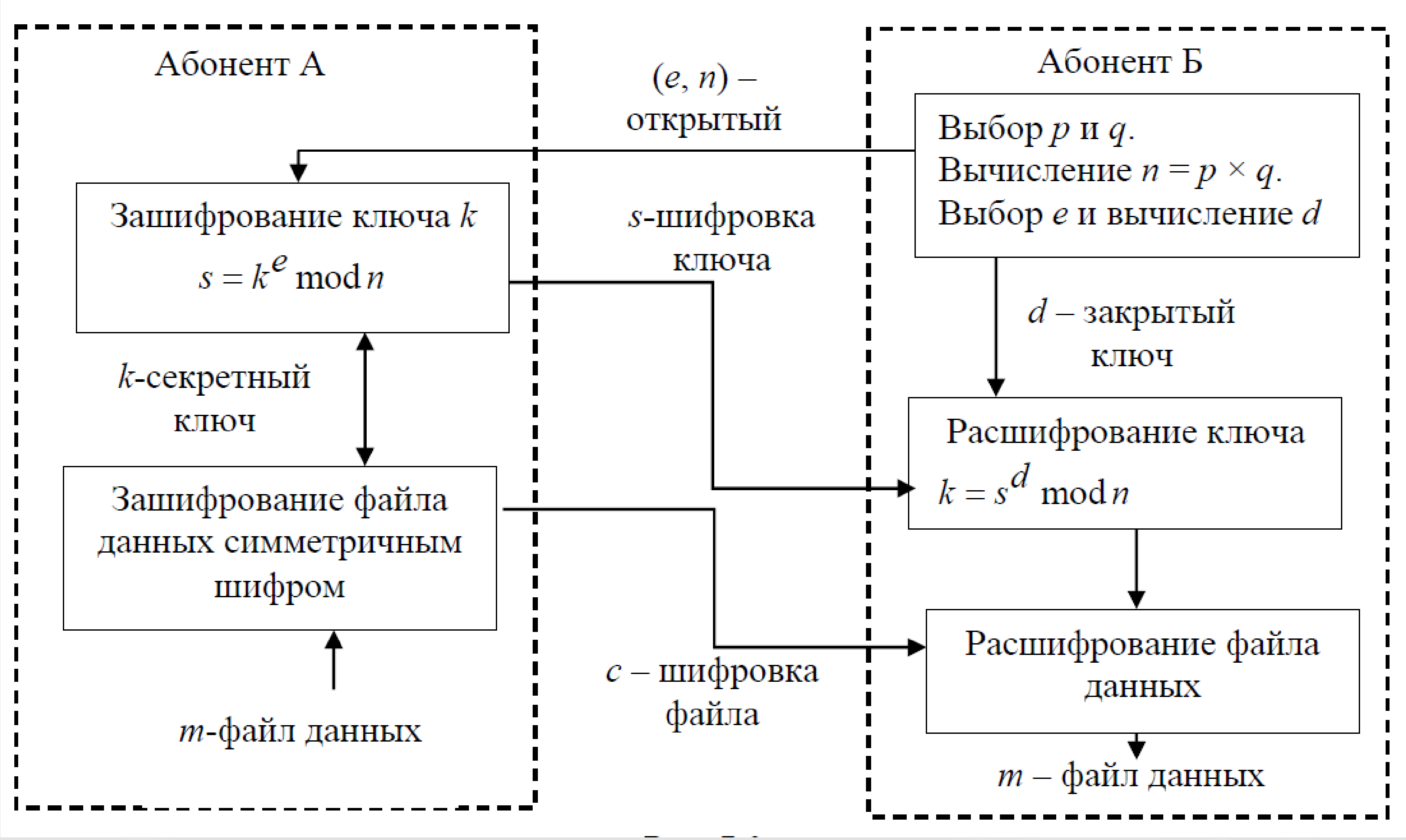
то находим его и выдаём результат: $x = \lambda S + x_2$

- Максимальное число умножений равно $2S \approx 2\sqrt{p-1}$ =2×2^{512}, что для практики очень велико
- Однако, так как числа $p - 1$ являются составными и если $p - 1$ можно разложить на маленькие множители, то криптоаналитик может применить процедуру, подобную процедуре Гельфонда, по взаимно простым делителям $p-1$ и найти секрет
- Наилучшие из известных решение задачи дискретного логарифмирования имеют экспоненциальную сложность порядка $O\left(e^{\sqrt{k}}\right)$, где $k = \lceil \log_2 p \rceil$ – битовая длина числа p.

Гибридное шифрования

Модель протокола гибридного шифрования

Пример гибридного шифрования на основе асимметричного шифра



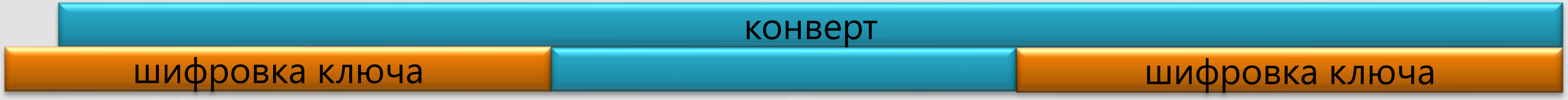
- Файл данных шифруется симметричным секретным ключом
- Секретный ключ шифруется открытым ключом получателя
- Зашифрованное сообщение и зашифрованный ключ составляют цифровой конверт (*digital envelope*), который отправляется получателю
- Получатель сначала расшифровывает секретный ключ, а затем расшифровывает секретным (сеансовым) ключом шифровку файла данных

Атака по побочным каналам на гибридную криптосистему

- Цель - определить симметричный секретный ключ, зашифрованный открытым ключом криптосистемы
- Условия атаки:
  - Нарушитель может перехватывать сообщения, адресованные серверу
  - Нарушитель может модифицировать сообщения и направлять их серверу
  - Сервер не определяет, от кого был получен конверт
  - Нарушитель может классифицировать ответы сервера на ПРИНЯТО/ОТКЛОНЕНО, т.е. случаи успешной и неуспешной расшифровки (по распознаванию ключевого слова)

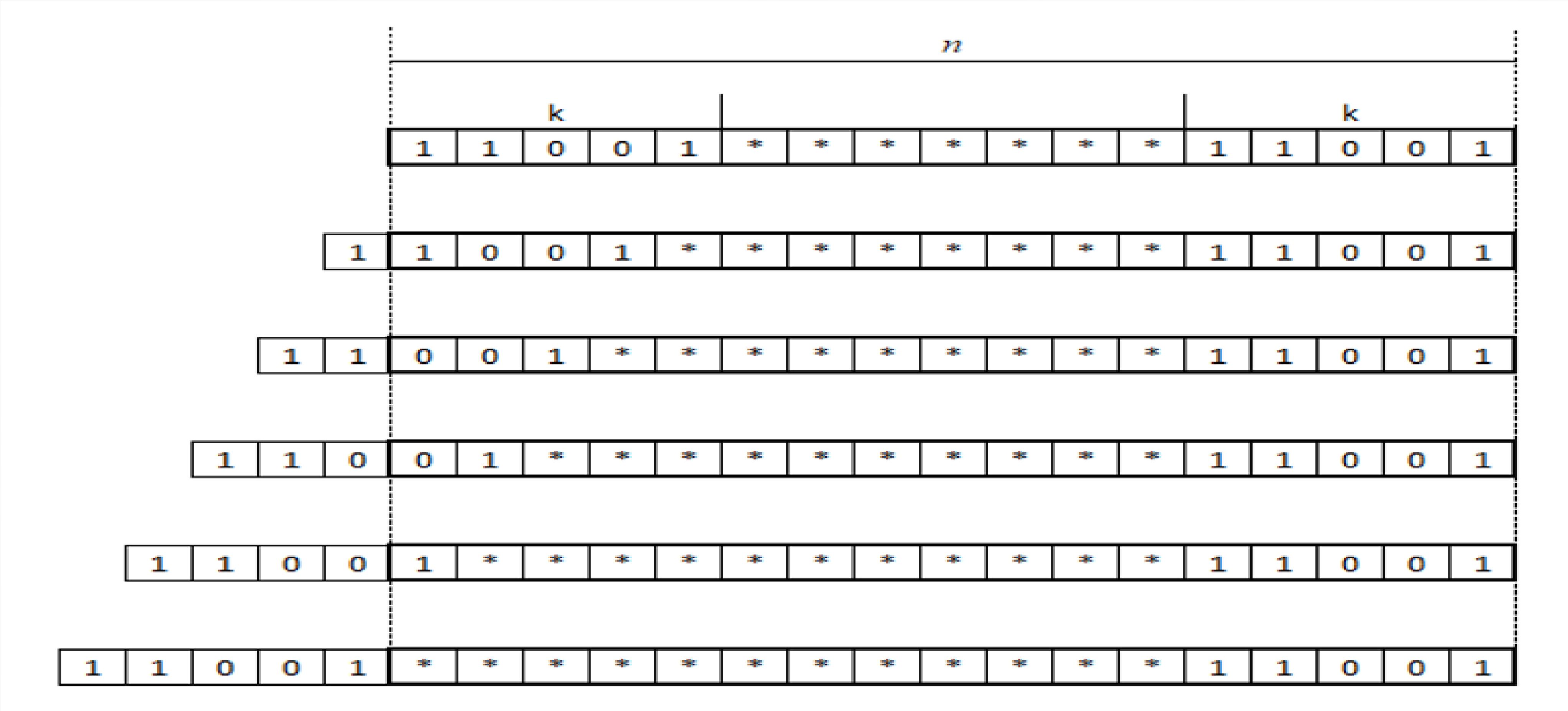
Идея атаки

- Длина в битах модуля n, используемого в RSA, существенно больше, чем длина в битах секретного ключа
- При расшифровке конверта сервер использует только младшие биты расшифрованного сообщения в качестве секретного ключа
- Модификация на первом шаге выполняется путем замены старших бит конверта шифровкой ключа, сдвинутой на один бит влево



- Анализируется ответ сервера: если ПРИНЯТО, то бит, следующий за старшим битом конверта – нулевой, а если ОТКЛОНЕНО то бит равен 1
- Продолжая действовать подобным образом, можно бит за битом восстановить целиком секретный ключ

Пример расшифровки модификаций ключа



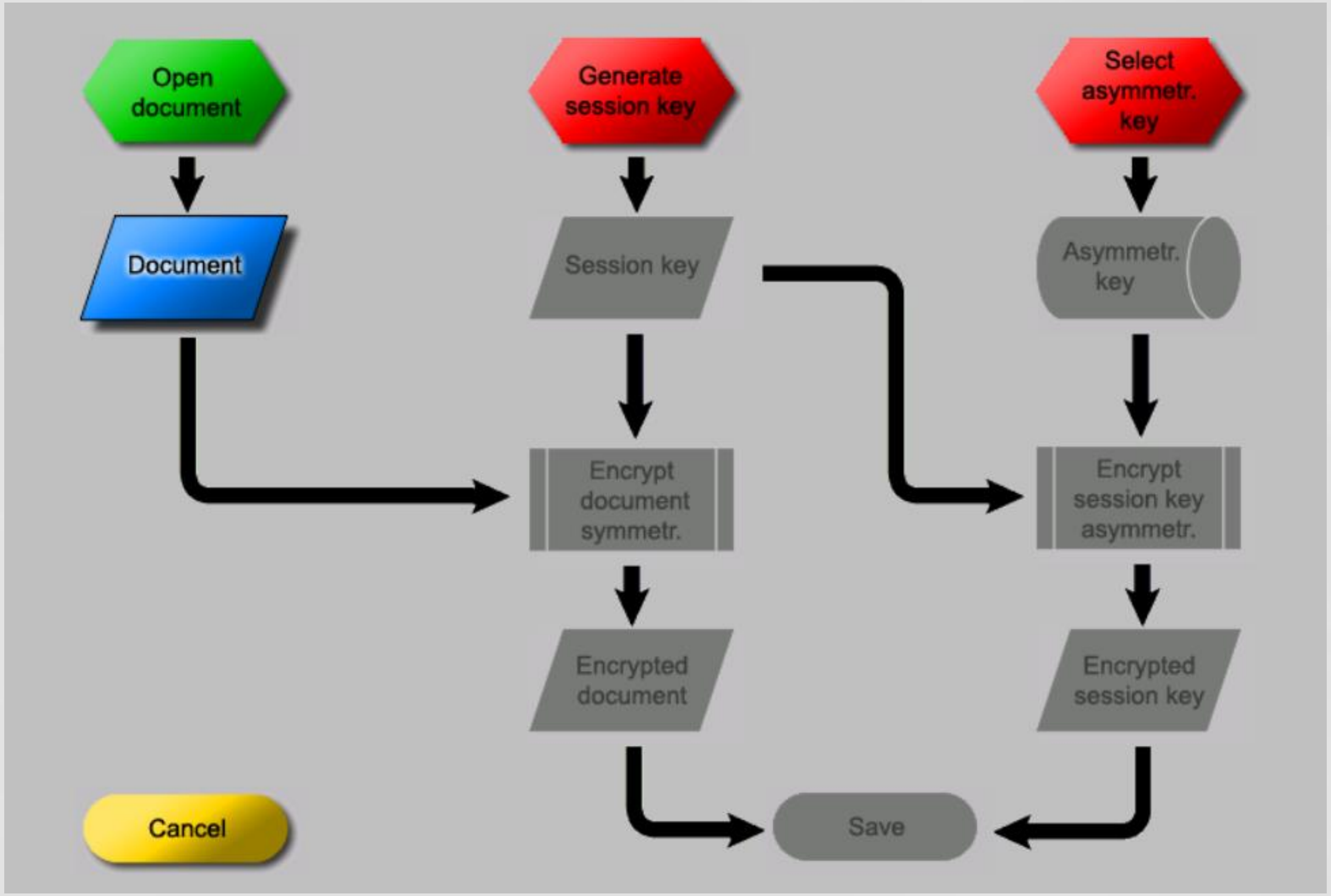
Модификация шифровки ключа:

$$K^e(1+2^l)^e mod\ N$$

Расшифровка модифицированного ключа:

$$K\left(1+2^l\right) mod\ N$$

Схема работы отправителя сообщения

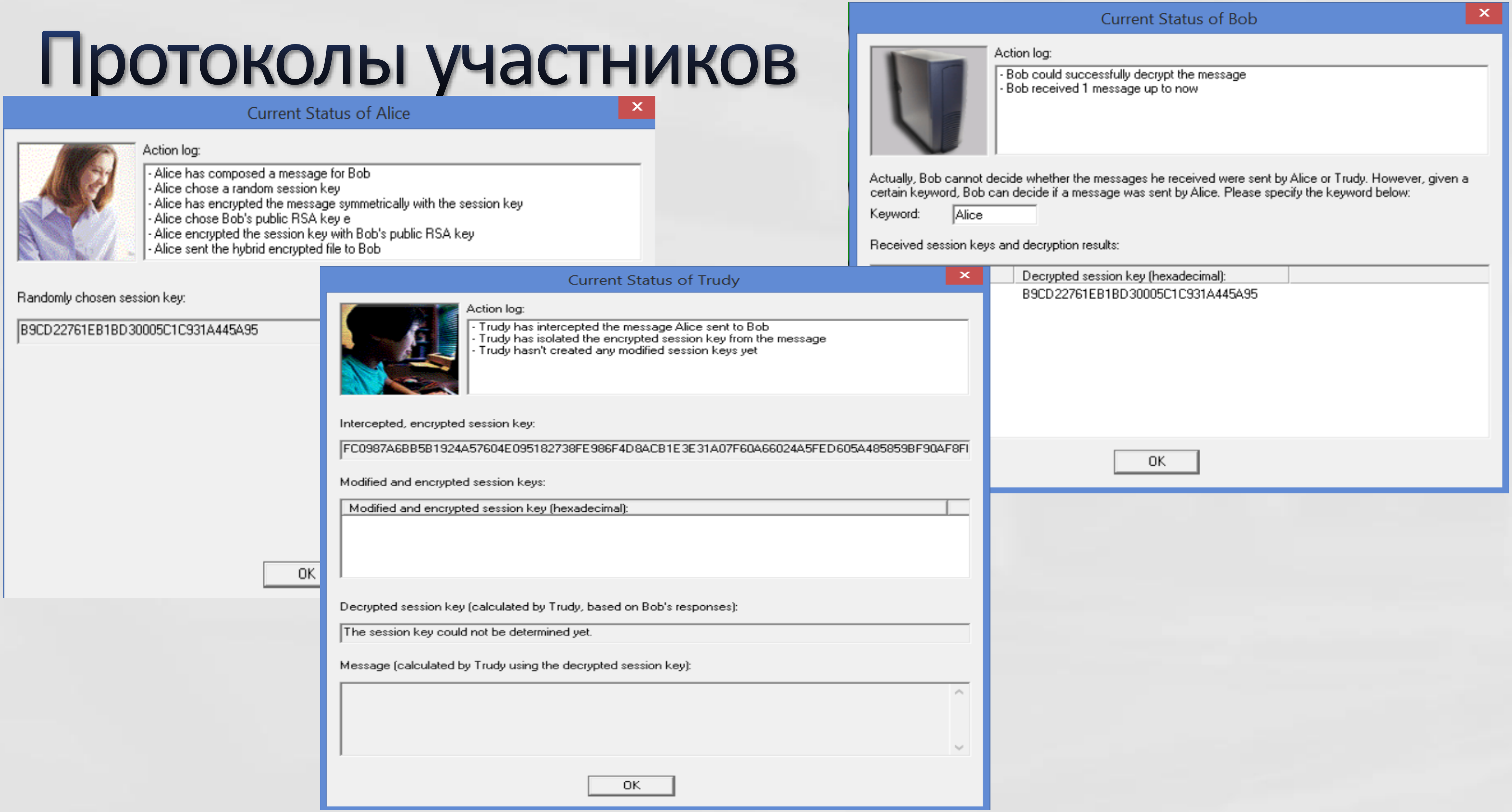


- «Зеленый» – выполненные действия
- «Красный» - действия, готовые к выполнению
- «Желтый» – завершающие действия
- «Серый» - действия не готовые к исполнению

Реализация атаки в Cryptool 1.0

- Действия участников протоколируются (доступ через )
- Порядок действий предопределен. «Красным» выделено очередное действие
- Полезна опция *«Show information dialogs»*

Протоколы участников



Приложение

Китайская теорема об остатках

- Пусть $n_{1,}n_{2,}\ldots n_{к}$ - натуральные попарно взаимно простые числа, а $r_{1,}r_{2,}\ldots r_k$ некоторые целые числа, тогда существует такое целое число М, которое является решением системы сравнений:
- $$\begin{cases} M \equiv r_1 mod n_1 \\ M \equiv r_2 mod n_2 \\ \qquad \ldots \\ M \equiv r_k mod n_k \end{cases}$$
- При этом для любых двух решений $A и B$ в этой системе справедливо $A \equiv B \bmod n_1 n_2 \ldots n_k$