Асимметричная криптография

Введение

Концептуальные отличия асимметричной криптографии

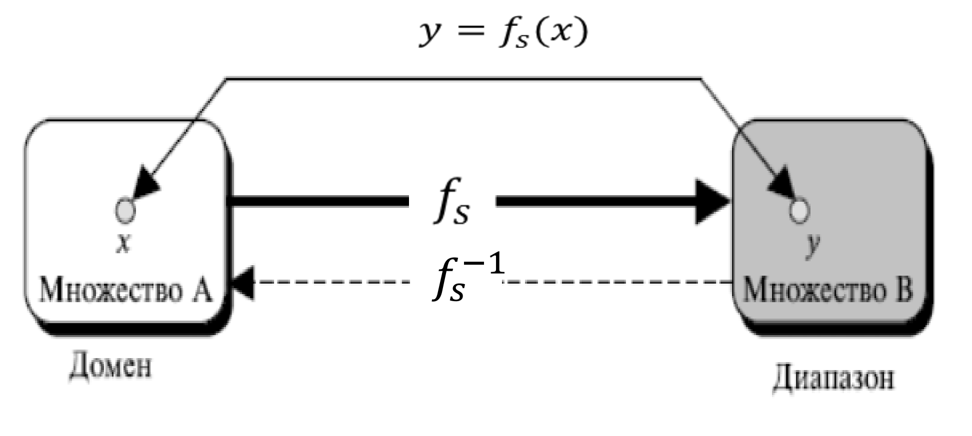
- Криптография с симметричными ключами базируется на совместном использовании секретного ключа, в то время как асимметричная криптография базируется на персональном ключе (закрытом)
- В криптографии с симметричными ключами, биты переставляются или заменяются другими; в асимметричной криптографии числа, представляющие открытые тексты, преобразуются с помощью математических функций

Историческая справка

- Первой открытой публикацией в области асимметричной криптографии принято считать статью Уитфилда Диффи (Whitfield Diffie) и Мартина Хеллмана (Martin Heilman) «Новые направления в криптографии», опубликованную в 1976 г.
- В «новой криптографии» введено понятие односторонней функции с секретом
- Предложен алгоритм, позволяющий паре пользователей выработать общий секретный ключ, не обмениваясь секретными данными по небезопасному каналу связи

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

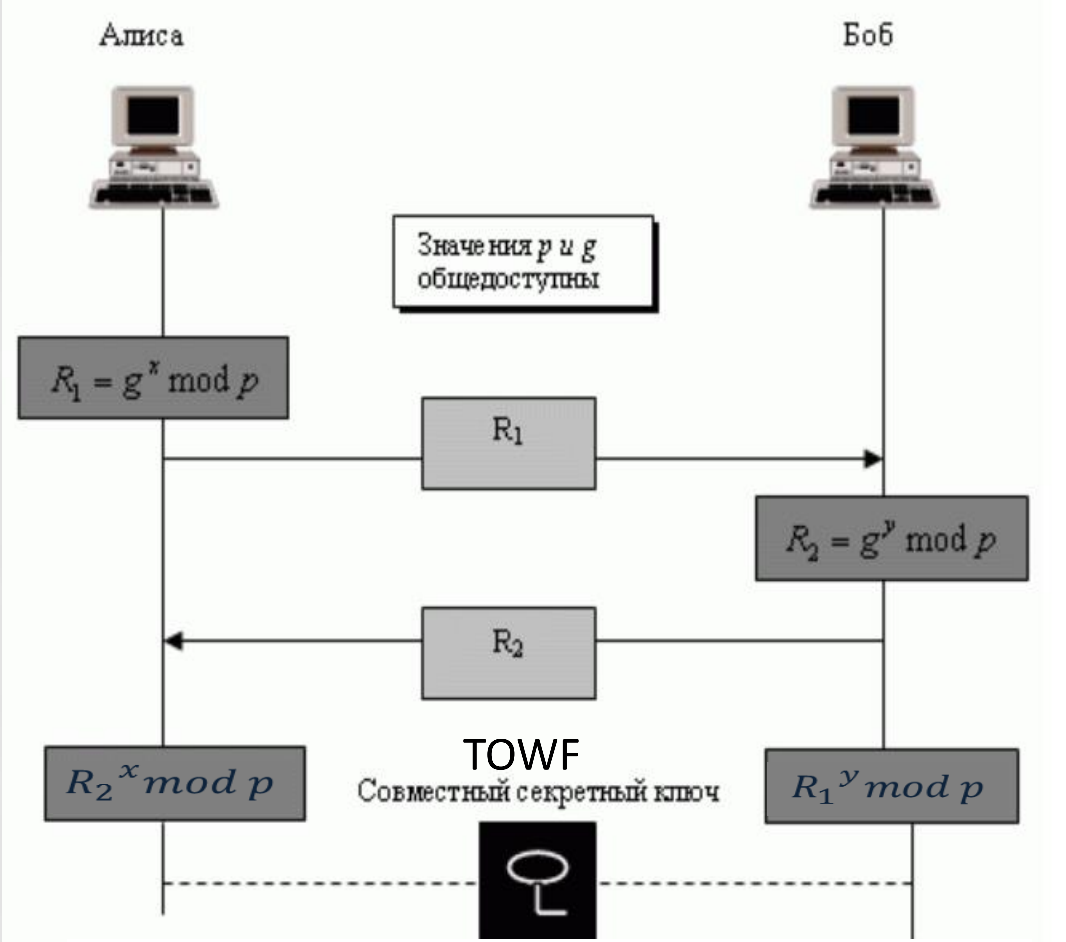
- Зная x, при любом s легко вычислить y=f_s(x)
- По известному значению y и s легко вычислить x=$f_s^{-1}(y)$
- Сложно вычислить x=$f_s^{-1}(y)$ по известному y, если секрет s не известен



Значимость TOWF

- Отказ от секретных каналов связи для предварительного обмена ключами;
- Включение в задачу вскрытия шифра трудную математическую задачу для повышения обоснованности стойкости шифра
- Решение новых криптографических задач, отличных от шифрования (электронная цифровая подпись и др.).

Протокол Диффи-Хеллмана (Diffie-Hellman, DH)



Симметричный общий ключ $\boldsymbol{K = g^{xy} mod\ p}$

- (p,g,R_1) и (p,g,R_2) - открытые ключи сторон
- x, y - закрытые ключи сторон
- $R_2^x mod\ p$ и $R_1^y mod\ p$ - односторонние функции с секретом (TOWF)

Математическая модель протокола

- *p* -большое простое число порядка 300 десятичных цифр (1024 бита)
- *g* – порождающий элемент циклической группы (генератор) порядка *p,* для которого справедливо: $g\ mod\ p, g^2 mod\ p, g^3 mod\ p \ldots g^{p-1} mod\ p$ являются различными целыми из *[1,p-1]*
- *x, y* - большие случайные числа такие, что *0 < x < p – 1, 0 < y < p – 1*
- Поскольку:

$$R_2{}^x mod\ p = (g^y mod\ p)^x mod\ p = g^{xy} mod\ p$$
$$R_1{}^y mod\ p = (g^x mod\ p)^y mod\ p = g^{xy} mod\ p$$

- Стороны фактически создают симметричный ключ сеанса без Центра распределения ключей (KDC)

Атака дискретного логарифма

- Так как *x* и *y* являются закрытыми данными, противник может получить только следующие значения g, p, R_1 , R_2
- Для вычисления ключа атакующий должен решить две задачи дискретного логарифмирования: найти целые *x* и *y* из уравнений $R_1 = g^x mod\ p; R_2 = g^y mod\ p$
- Задача вычисления дискретные логарифмов становится трудноразрешимой, если:
  - Простое число *p* должно быть очень большим (более чем 300 десятичных цифр).
  - Простое число *p* должно быть выбрано так, чтобы *p - 1* имел по крайней мере один простой делитель (больше чем 60 десятичных цифр).
  - Генератор должен *g* быть первообразный корень по модулю *p*
  - Значения *x* и *y* должны использоваться только единожды

Пример

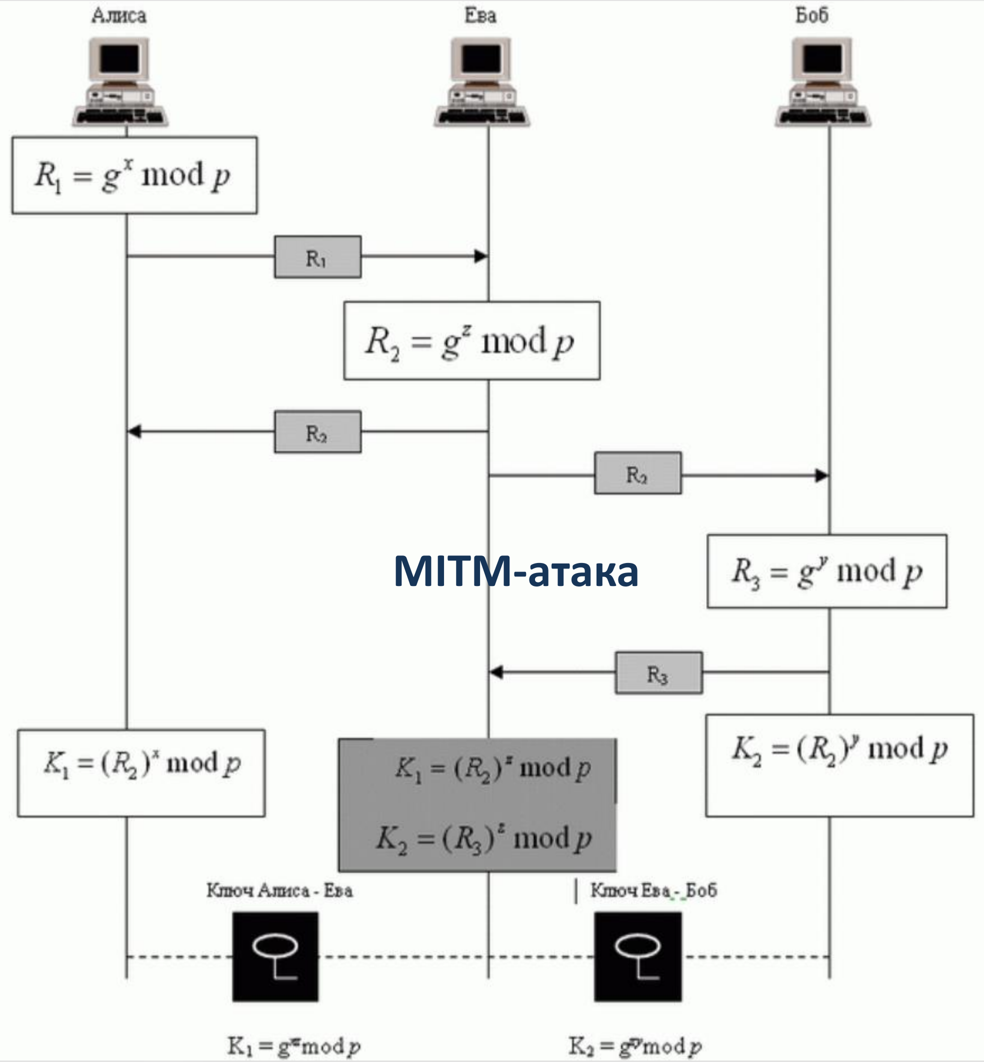
- Выбираем:

p	764624298563493572182493765955030507476338096726949748923573772860925 2356666607554236374233096611800333381061947301309504147387009991780436548785807987581
g	2
x	557
y	273

- Вычисляем:

R_1	84492028420 66550521617294749103509414343369852001266086286363106767 3619959280828586700802131859290945140217500319973312945836083821943065 966020157955354
R_2	4352628387092003794707471148955816276363891162621155579751233792185 66 310011435718208390040181876486841753831165342691630263421106721508589 6255201288594143
K	15563800066452229059622582752327076527321804694442367852032040014640 6 500887936651204257426776608327911017153038674561252213151610976584200 1204086433617740

Атака посредника (man in the middle)



- Предполагается, что противник может осуществить активную атаку, т.е. имеет возможность не только перехватывать сообщения, но и заменять их другими
- Противник может перехватить открытые ключи участников R_1 и R_2 и создать свою пару открытого и закрытого числа (R_3, z), чтобы послать их каждому из абонентов
- После этого каждый абонент вычислит ключ, который будет общим с противником, а не с другим участником
- Если нет контроля подлинности сторон, то законные абоненты не смогут обнаружить подобную подмену

Обучающий ролик по DH-протоколу

https://www.youtube.com/watch?v=vFjq9pID4-E

Модель шифрования в асимметричной криптосистеме

Свойства асимметричной криптосистемы

- В асимметричных системах 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)

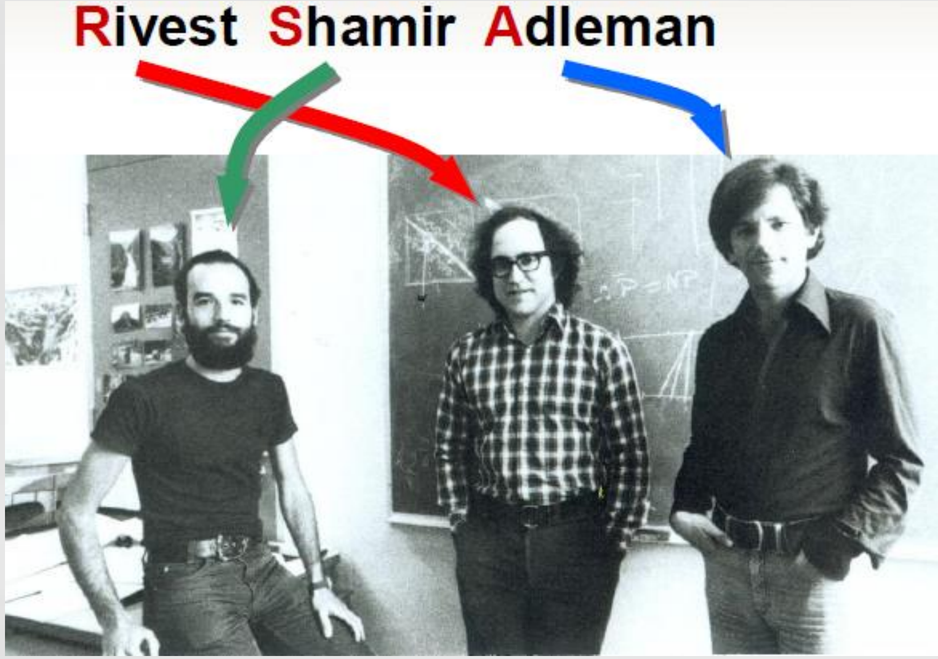
- Вычислительно легко создавать пару (открытый ключ, закрытый ключ)
- Вычислительно легко, имея открытый ключ и незашифрованное сообщение, создать соответствующее зашифрованное сообщение
- Вычислительно легко расшифровать сообщение, используя закрытый ключ
- <u>Вычислительно невозможно, зная открытый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

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 *с 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

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

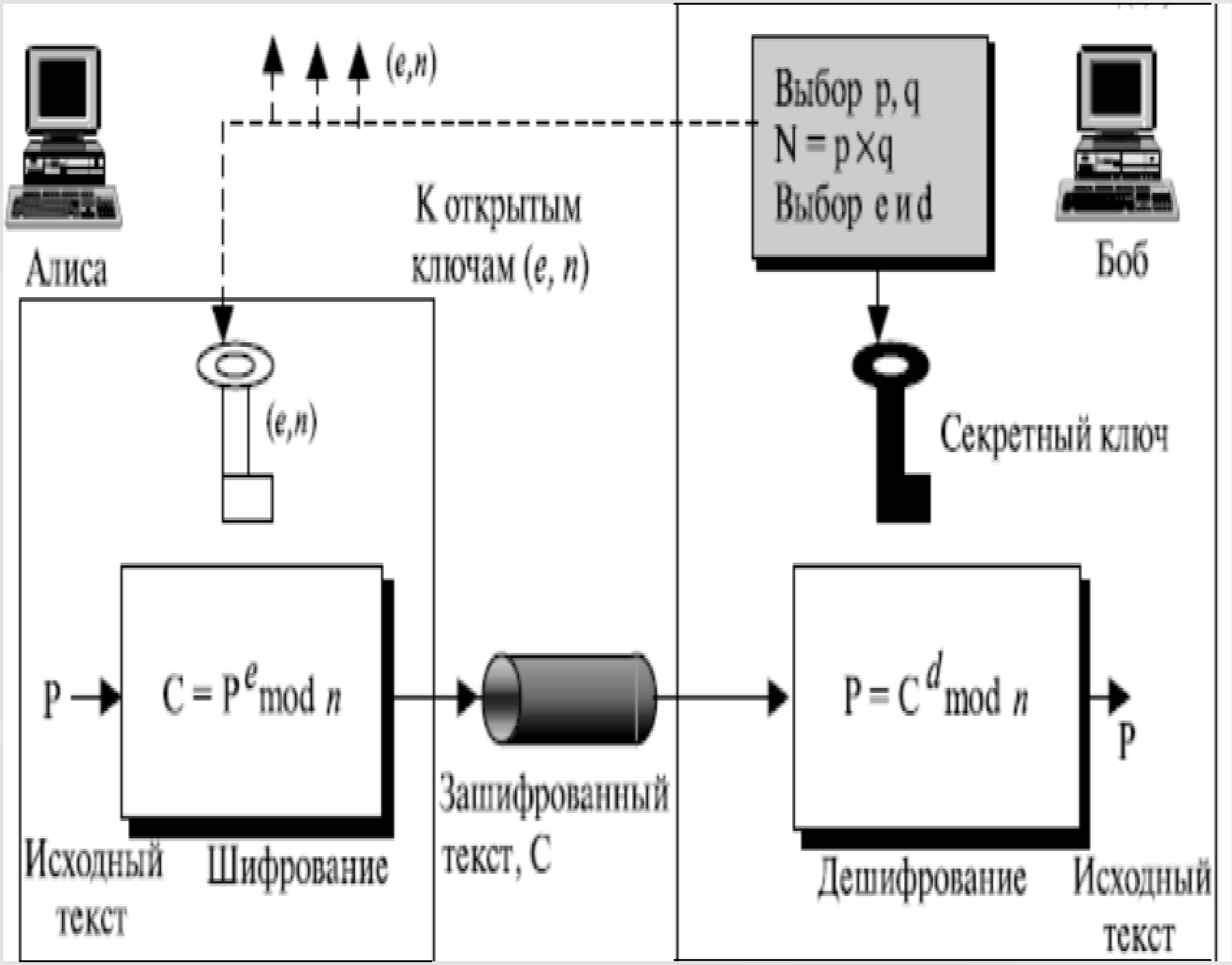
умножим на x левую и правые части. В итоге получаем:

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

- 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 конфиденциальной передачи сообщения 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=$a^{((\dots((m_k*2+m_{k-1})*2+m_{k-2})*2+\cdots)*2+m_1)*2+m_0}$=
- $((\dots(((a^{m_k})^2 * a^{m_{k-1}})^2 \dots)^2 * a^{m_1})^2 a^{m_0}$
- Получаем мультипликативный аналог схемы Горнера:
- $$\begin{cases} s_1 = a\, mod\, n \\ s_{i+1} = {s_i}^2 \quad *a^{m_{k-i}} \text{mod n} \\ i = 1, \dots, k \end{cases}$$
- 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 E 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 A
- **Противодействие - каждый абонент должен использовать свой собственный модуль**

Атака с выборкой зашифрованного текста

- **Цель- получение открытого текста сообщения**
- Нарушитель Е перехватывает шифровку $C = P^e mod\ n$ для получателя В
- Нарушитель имеет возможность обманом («ослепление») получить от В 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

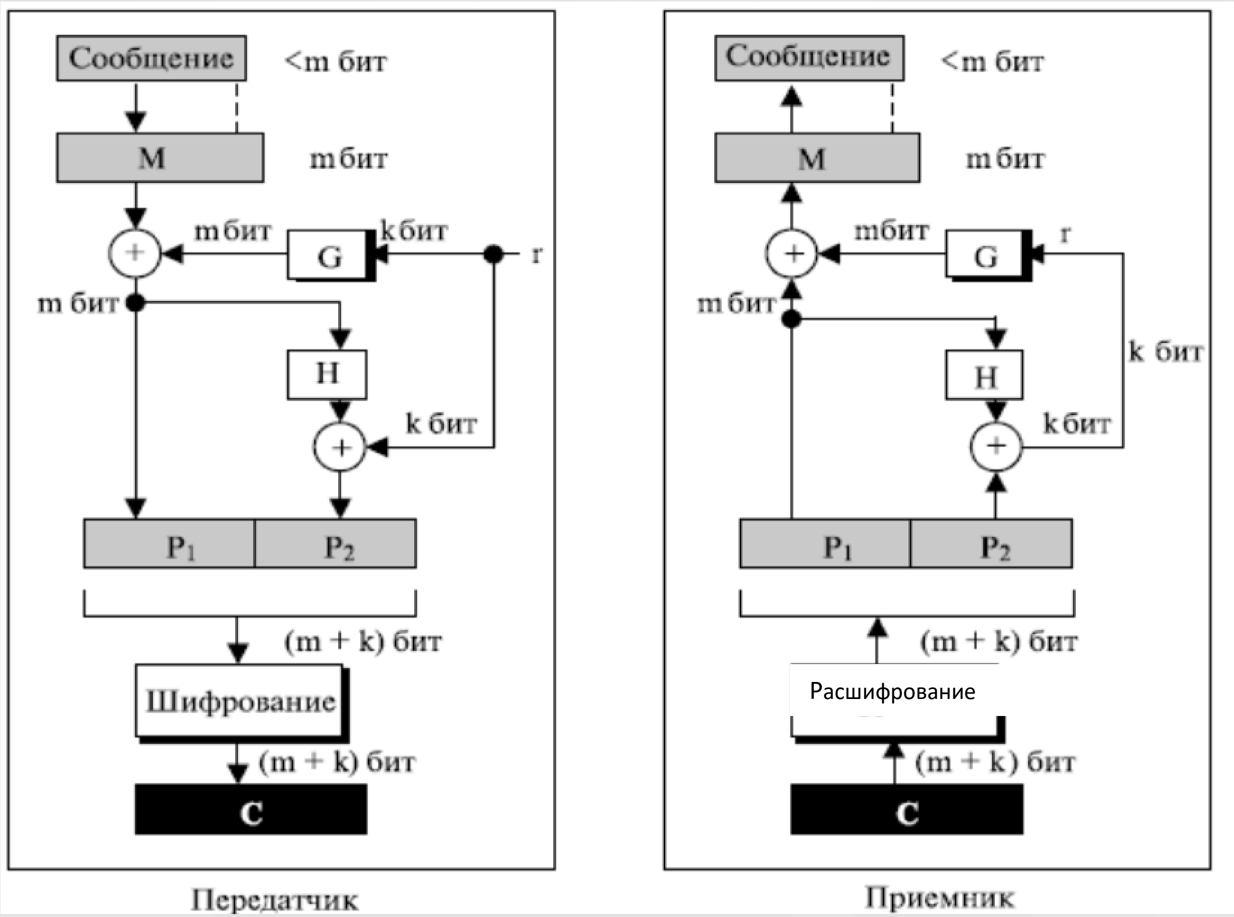
Атака при малом показатели степени (ключе) шифрования

- **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 Assimetric 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

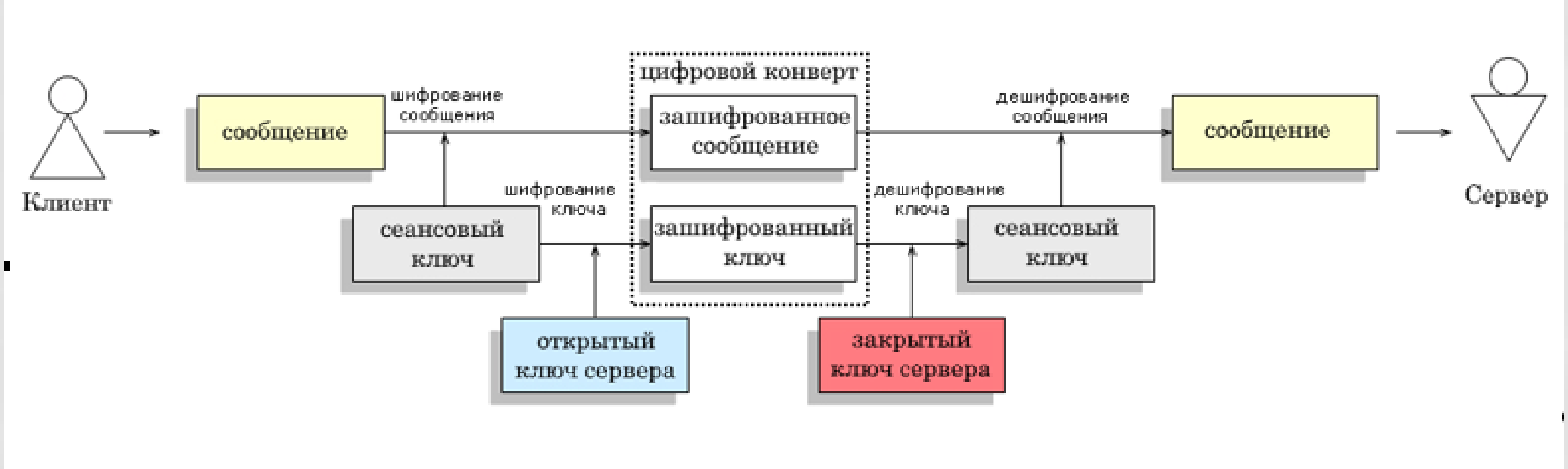
- Число битов для n должно быть, по крайней мере, 1024. Это означает, что n должно быть приблизительно 2^{1024} или 309 десятичных цифр
- Два простых числа p и q должны каждый быть по крайней мере 512 битов. Это означает, что p и q должны быть приблизительно 2^{512} или 154 десятичными цифрами
- Значения 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.)

Модель шифрования в гибридной криптосистеме

Гибридная криптосистема на основе асимметричного шифра



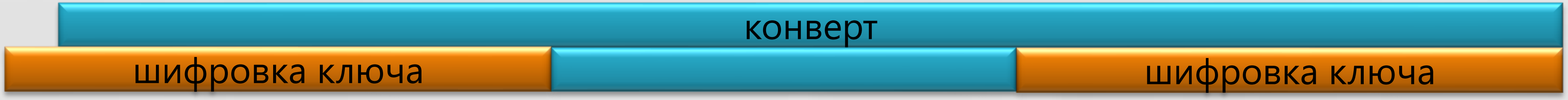
- Сообщение шифруется симметричным секретным ключом
- Секретный ключ шифруется открытым ключом получателя
- Зашифрованное сообщение и зашифрованный ключ составляют цифровой конверт (*digital envelope*), который отправляется получателю
- Получатель сначала расшифровывает секретный ключ, а затем расшифровывает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

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)

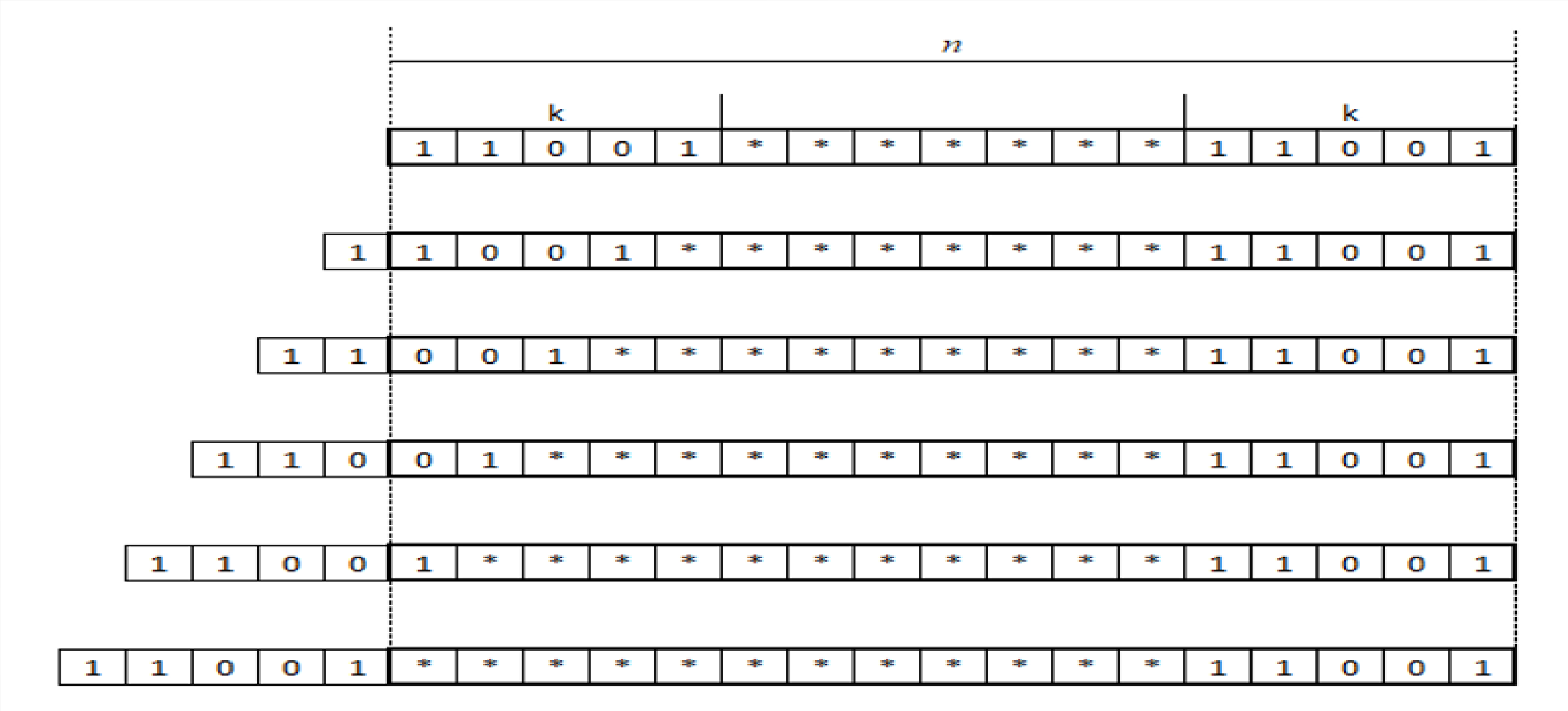
Идея атаки

- 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



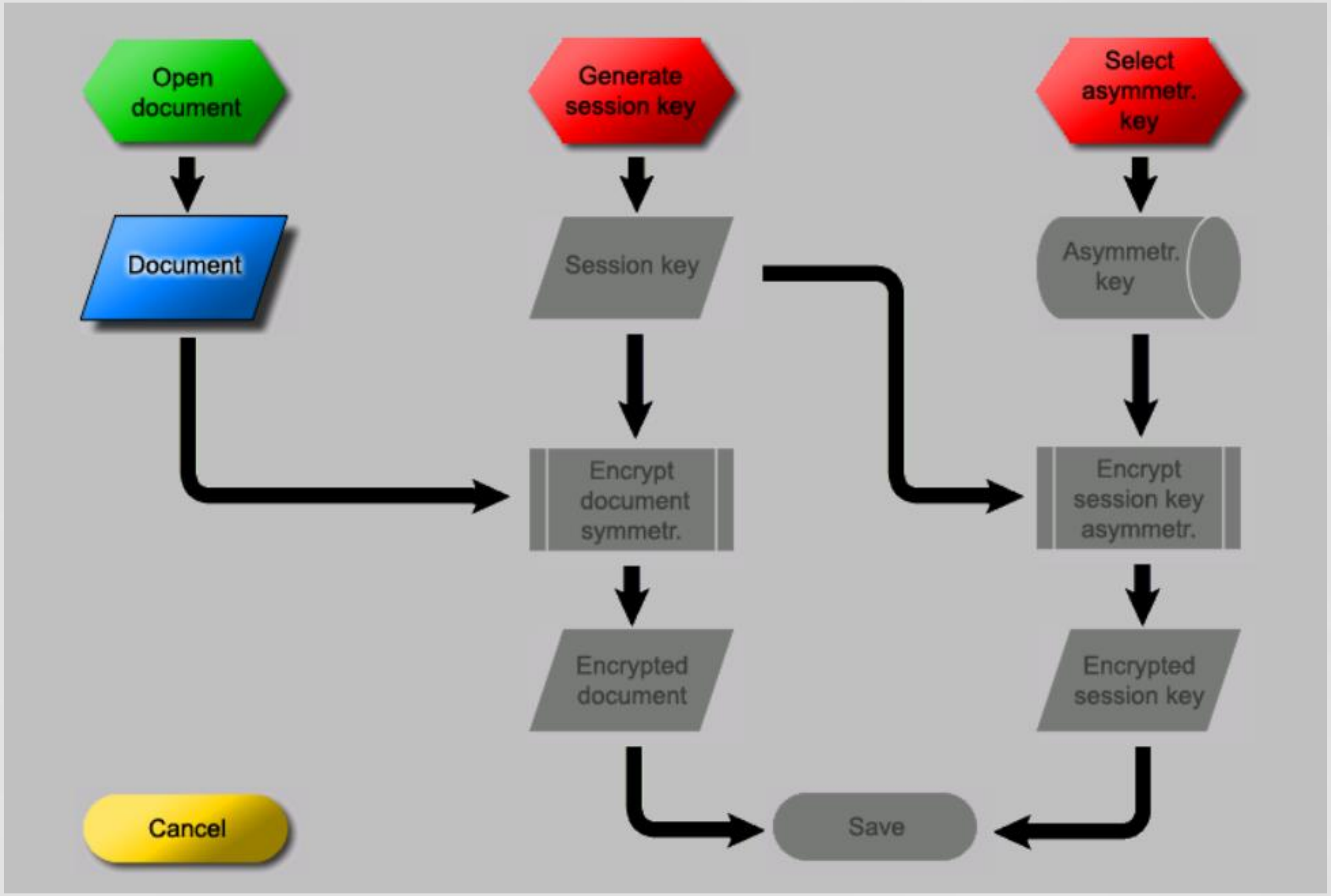
Модификация шифровки ключа:

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

Расшифровка модифицированного ключа:

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

Схема работы отправителя сообщения

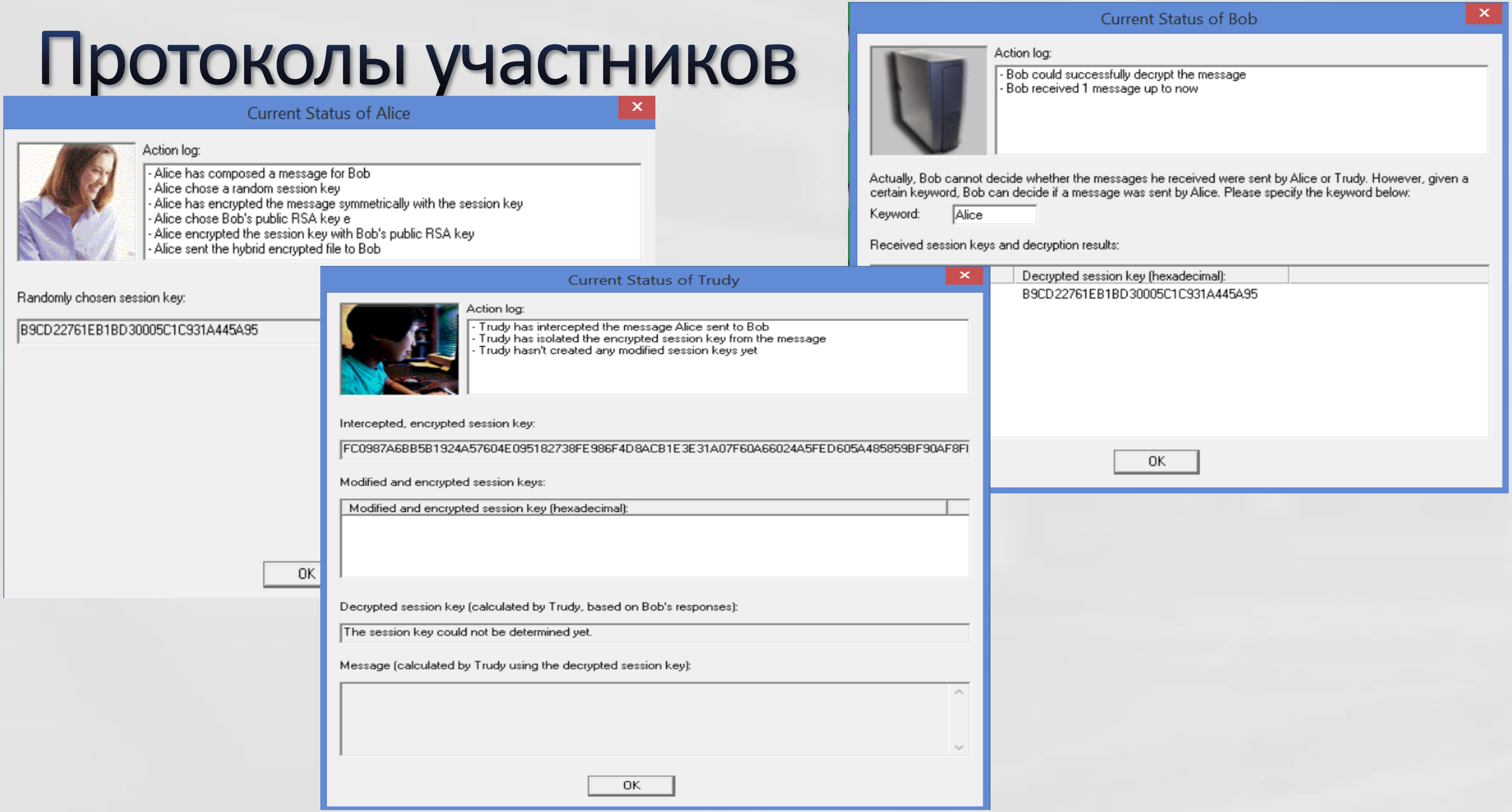


- «Зеленый» – выполненные действия
- «Красный» - действия, готовые к выполнению
- «Желтый» – завершающие действия
- «Серый» - действия не готовые к исполнению

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:
- $$\begin{cases} M \equiv r_1 mod n_1 \\ M \equiv r_2 mod n_2 \\ \quad\quad \ldots \\ M \equiv r_k mod n_k \end{cases}$$
- При этом для любых двух решений $A и B$ в этой системе справедливо $A \equiv B \bmod n_1 n_2 \ldots n_k$