# Лекция 7. Асимметричное шифрование

## Идея асимметричного шифрования

Есть проблема распространения ключей.

Для шифрования и расшифровки используются разные ключи, один из которых можно открыто передавать.

Используется обратимая математическая функция, легко вычисляемая в одном направлении, но сложно в обратном (RSA, Эль-Гамаль, ГОСТ).

M

Encrypt (k1, M)

C=E(M)

C

Decrypt (k2, C)

D(C)=D(E(M))=M

1976 - идею предложили Уитфилд Диффи (Whitfield Diffie) и Мартин Хеллман (Martin Hellman), и независимо Ральф Чарльз Меркл (Ralph Charles Merkle):

- алгоритм общедоступен

- ключ шифрования общедоступен (распространяется публично)

- ключ расшифровки - секрет

- обратное преобразование с помощью открытого ключа сложно

Снимается требование личной встречи для формирования секретного ключа.

1973 - Джеймс Эллис, **Клиффорд Кокс** и М.Д. Уильямсон изобрели то же самое, но работали на британскую разведку (Правительственная связь) и получили разрешение на публикацию только в 1997 году.

## Элиптическая криптография

1985 - независимо Нил Коблиц и Виктор Миллер

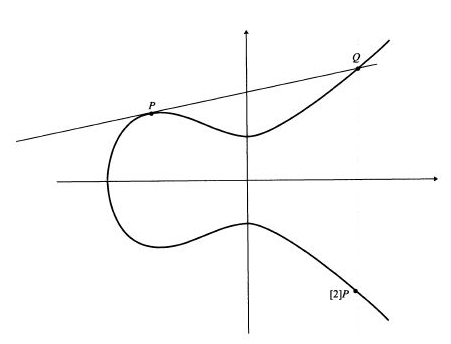
Использовать в криптографии алгебраические свойства эллиптических кривых( "криптография на эллиптических кривых")

Основная криптографическая операция - скалярное умножение точки на эллиптической кривой на целое число. Определяется через сложение и удвоения точек эллиптической кривой (выполняются на основе операций сложения, умножения и инвертирования в конечном поле, над которыми рассматривается кривая).

Безопасность: не существует субэкспоненциальных алгоритмов дискретного логарифмирования в группах точек кривой (порядок группы точек эллиптической кривой определяет сложность задачи). Для достижения уровня криптостойкости как и в RSA, требуются группы меньших порядков.

На конференции RSA 2005 Агентство национальной безопасности объявило о создании “Suite B”, в котором используются исключительно алгоритмы эллиптической криптографии, причём для защиты информации, классифицируемой до “Top Secret”, используются всего лишь 384-битные ключи.

ГОСТ 34.10-2012 (ЭП на эллиптических кривых)



## RSA

Назван в честь его изобретателей Рона Ривеста, Ади Шамира и Леонарда Элдмана (Rivest/Shamir/Adleman) из Массачусетского Технологического Института. Впервые описание алгоритма RSA было опубликовано в августе 1977 года в журнале Scientific American.

**Основание:** предположение о сложности разложения целых чисел на множители.

Два числа легко перемножить и найти произведение. Но если дано только произведение, практически невозможно разложить число на множители и определить исходные числа.

1977 - в колонке «Математические игры» Мартина Гарднера в Scientific American, с разрешения Рональда Ривеста опубликовано первое описание криптосистемы RSA.

Читателям было предложено дешифровать английскую фразу, зашифрованную описанным алгоритмом с открытыми параметрами n=1143816...6879541 (129 десятичных знаков, 425 бит, также известно как RSA-129) и e=9007. Награда за расшифровку - 100 долларов США.

Ривест: для факторизации числа потребовалось бы более 40 квадриллионов лет.

3 сентября 1993 года - стартовал проект распределённых вычислений с координацией через электронную почту по нахождению сомножителей числа RSA-129.

220 дней 600+ добровольцев из 20 стран жертвовали процессорное время 1600 машин.

В результате найдены простые множители и расшифровано исходное сообщение: «THE MAGIC WORDS ARE SQUEAMISH OSSIFRAGE («Волшебные слова — это брезгливый ягнятник»). Полученную награду победители пожертвовали в Free Software Foundation.

[Схема Эль-Гамаля](https://ru.wikipedia.org/wiki/%D0%A1%D1%85%D0%B5%D0%BC%D0%B0_%D0%AD%D0%BB%D1%8C-%D0%93%D0%B0%D0%BC%D0%B0%D0%BB%D1%8F" \o "Схема Эль-Гамаля)

#### Шифрование RSA

**Вычисление ключей**

Ключи создаются вместе, так как они математически зависимы.

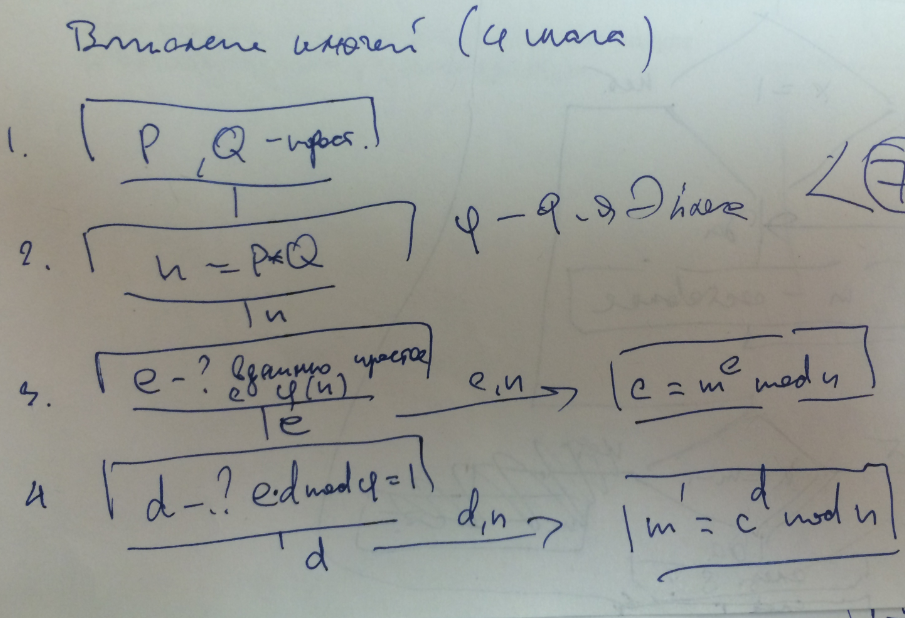
Открытый и закрытый ключи являются функциями двух больших случайных простых чисел p и q, для которых рассчитывается произведение n=p\*q. Число n называется модулем шифрования, так как определяет степень «размывания» области шифруемых значений по множеству зашифрованных значений.

Затем выбирается случайное число e, такое что он и значение выражения (p-1)\*(q-1) являются взаимно простыми числами (то есть такими числами, которые имеют единственный общий делитель – единицу). Для поиска числа e может использоваться алгоритм Эвклида, позволяющий найти наибольший общий делитель для двух чисел.

Найденное число e считается вместе с числом n открытым ключом.

После определения числа e вычисляется такое число d, что выполняется условие (e\*d)mod((p-1)\*(q-1))=1. Для нахождения решения данного уравнения используют расширенный алгоритм Эвклида.

Следует отметить, что d и n также являются взаимно простыми числами.



**Шифрование/расшифровка сообщения**

Для шифрования сообщения оно разбивается на блоки, длина которых равна максимальной степени числа 2, но не превышает n (то есть, для 32 разрядных p и q и, соответственно, 64 разрядного n, шифруемый блок должен определяться как max(2^64)<n).

Если размер блока оказывается меньше заданного значения, перед шифрованием он может быть дополнен любой информацией до нужного размера. При этом необходимо сохранить информацию о количестве значимой информации в данном блоке для восстановления исходной информации после расшифровки.

Зашифрованный блок получается из исходного по формуле:

c(i)=(m(i)^e)mod(n)

Для расшифровки блока используется формула:

m(i)=(c(i)^d)mod(n)

Справедливость обратного преобразования определяется следующим выводом (все по mod(n)):

c(i)^d=(m(i)^e)^d=m(i)^(e\*d)=m(i)^(1+k\*(p-1)\*(q-1))=(m(i)^1)(**m(i)^(k\*(p-1)\*(q-1))**)= m(i)\*1= m(i)

**Функция Эйлера и Теорема Эйлера**

Фу́нкция Э́йлера φ ( n ) - мультипликативная арифметическая функция, равная количеству натуральных чисел, меньших n и взаимно простых с ним. По определению число 1 взаимно просто со всеми натуральными числами, и φ (1) = 1.

Например, для числа 24 существует 8 меньших его и взаимно простых с ним чисел (1, 5, 7, 11, 13, 17, 19, 23), поэтому φ (24) = 8.

Теорема Эйлера: а^ φ (n)=1

**Пример вычисления ключа и шифрования/расшифровки**

Пусть p=17, q=19, тогда n=323. Открытый ключ e не должен иметь общих множителей с числом (p-1)\*(q-1)=16\*18=288, и может быть равен, например 35. Тогда закрытый ключ d можно получить из уравнения (35\*d)mod(288)=1 (нас интересует k\*фи – это k\*288 - заканчивающееся на 4 или 9), то есть одним из значений d (при k=13 и k\*фи=3744) будет 107.

Открытый ключ (e=35 вместе с модулем n=323) передается всем, кто собирается направлять зашифрованные сообщения.

Закрытый ключ хранится в секрете (исходные простые числа аккуратно уничтожаются).

Для шифрования сообщения m=1234567 разделим его на блоки по 8 бит. получим блоки m(1)=123, m(2)=45, m(3)=67.

После возведения в степень e=35 зашифрованные блоки будут следующие: c(1)=, c(2)=, c(3)=.

Для расшифровки требуется возведение в степень d=107 по модулю n=323.

**Безопасность алгоритма RSA**

Основана на трудности разложения на множители больших чисел. А так как открытый и закрытый ключи являются функциями двух простых чисел, восстановление текста по зашифрованному тексту и открытому ключу эквивалентно разложению на множители двух больших чисел.

Для повышения безопасности ключей **простые числа p и q должны иметь равную длину**. Важно, чтобы эти числа не были раскрыты.

**Основной недостаток алгоритма RSA**

Сложность выполняемых вычислений, что не позволяет достичь высокой скорости шифрования (в 1000 раз медленнее DES). Алгоритм не может использоваться для поточного шифрования в системах, критически относящимся к задержкам реакции.

Для существенного ускорения вычислений используют специально подобранные значения для числа e: 3, 17, 257 или 65537(2^16+1 – это 17 умножений), в двоичном представлении которых взведены только 2 бита. Это позволяет при шифровании сократить число умножений до разрядности самого числа e.

**Применение алгоритма RSA**

Алгоритм шифрования RSA используется для защиты программного обеспечения и в схемах цифровой подписи.

Также она используется в открытой системе шифрования PGP и иных системах шифрования (к примеру, DarkCryptTC и формат xdc) в сочетании с симметричными алгоритмами.

**Гибридная криптосистема**

Из-за низкой скорости шифрования (около 30 кбит/с при 512 битном ключе на процессоре 2 ГГц), сообщения обычно шифруют с помощью более производительных **симметричных алгоритмов со случайным ключом (сеансовый ключ)**, а с помощью RSA шифруют лишь этот ключ.

Такой механизм имеет потенциальные уязвимости ввиду необходимости использовать криптостойкий генератор случайных чисел для **формирования случайного сеансового ключа** симметричного шифрования и эффективно противостоящий атакам симметричный криптоалгоритм (в данное время широкое применение находят AES, IDEA, Serpent, Twofish).

## Получение простых чисел

#### Решето Эратосфена

Остальные алгоритмы получают случайное число и проверяют его на простоту.

#### Алгоритм, следующий из определения простого числа

Простое число N – делится нацело только на само себя и на единицу.

Проверка: делится ли это число на все числа интервала [2; N-1]. Если число N не разделится ни на одно число из приведенного ряда, оно простое.

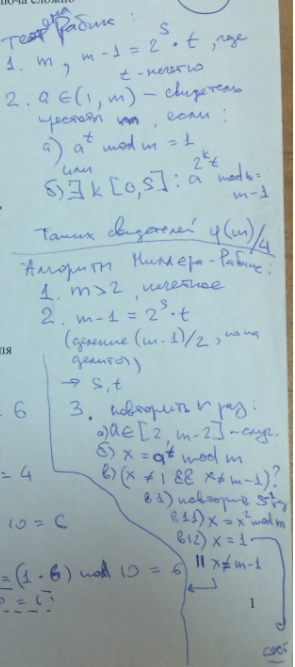
Привлекательность данного метода омрачается лишь его сложностью.

Единственным усовершенствованием приведенного алгоритма было доказательство теоремы о том, что все делители числа лежат в диапазоне интервале [2; SQRT(N)], что несколько сократило число операций деления, необходимых для проверки числа на простоту.

Все равно очень медленный, поэтому используют специальные алгоритмы определения простоты числа.

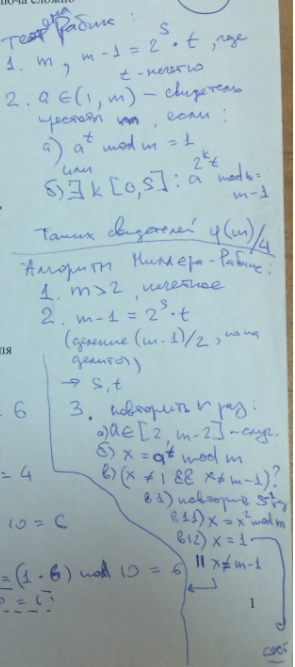
#### Теорема Рабина

**<<схема>>**



#### Тест Рабина-Миллера

**<<схема>>**



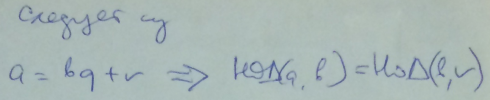
Проверка взаимной простоты чисел (числа, имеющие НОД=1)

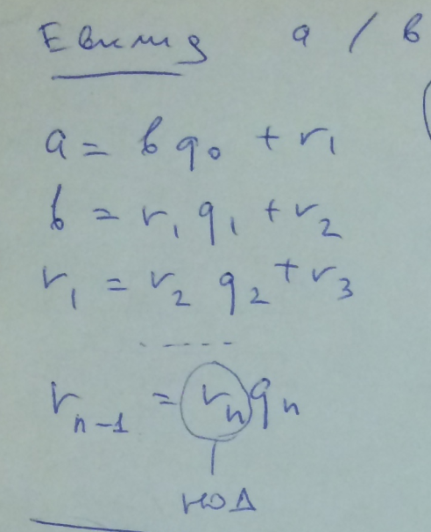
#### Алгоритм Евклида

Общая мера двух отрезков - 325 г. до н.э.

Аристотель – 384 г. до н.э.

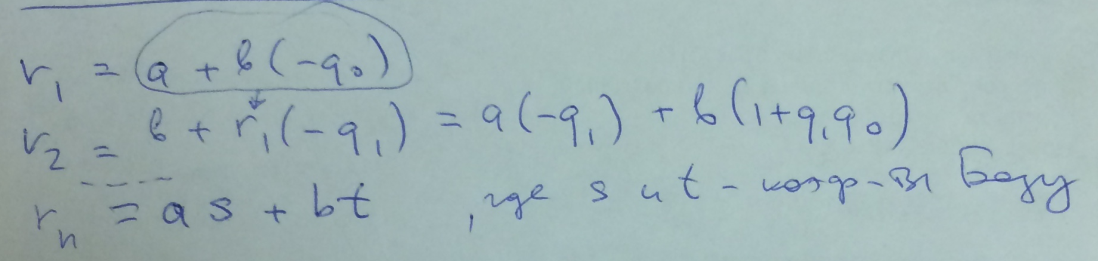
НОД (a, b) = НОД (b, a/b)

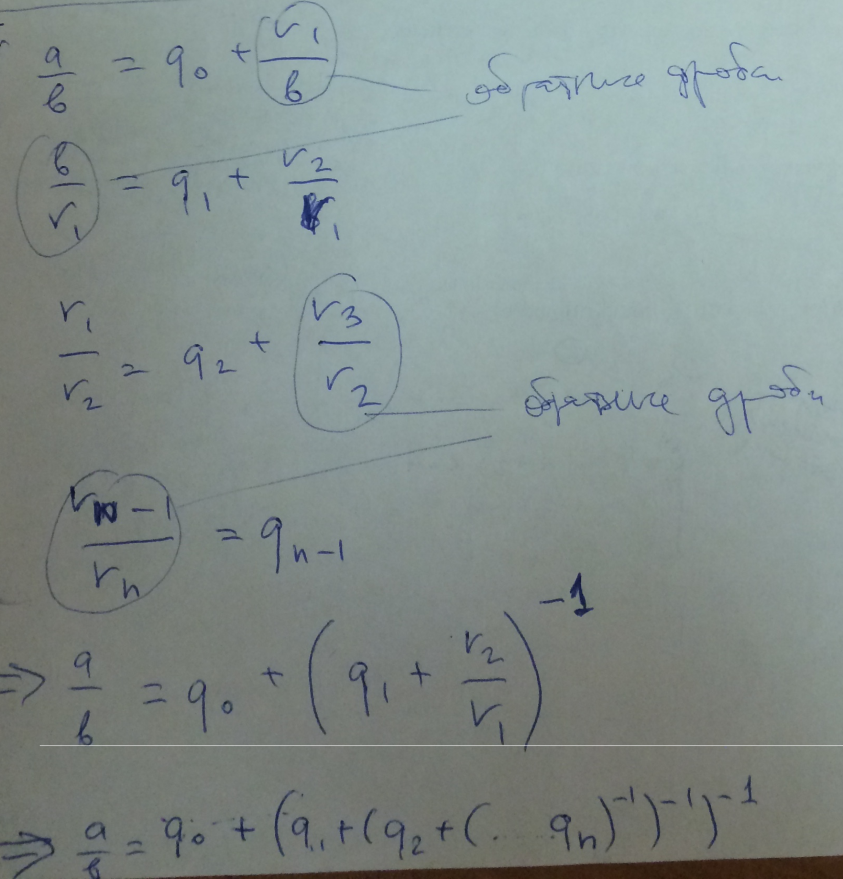




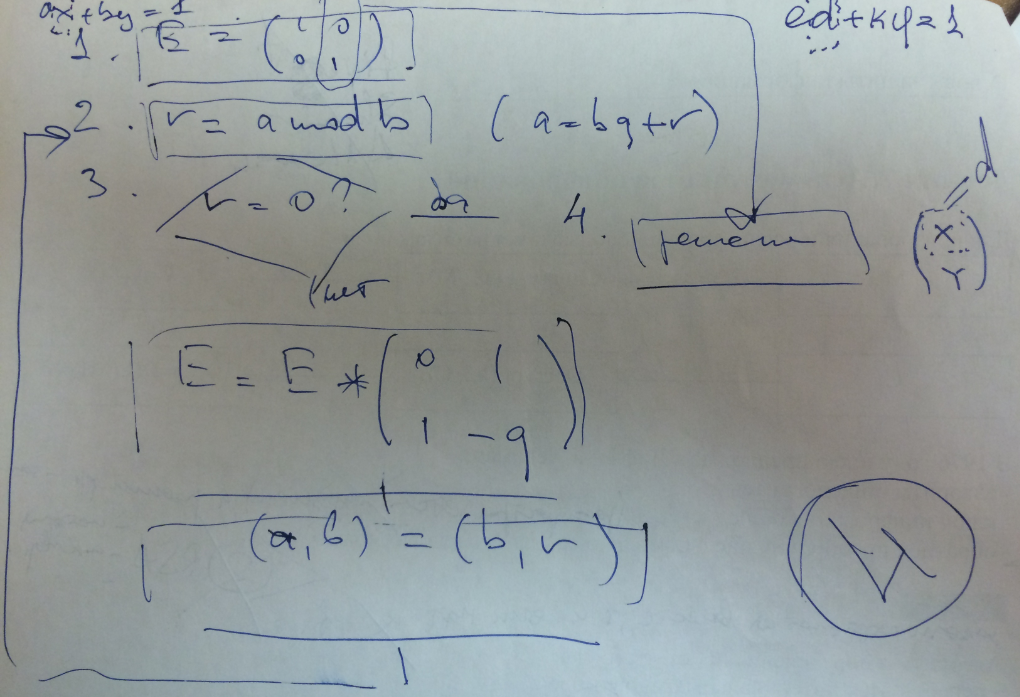
#### Расширенный алгоритм Евклида

Через коэффициенты





Матричный



#### Алгоритм быстрого возведения в большую степень

Операция возведения в степень требует использования переменных огромной размерности. Сама операция при этом может потребовать значительного времени. Для устранения данного недостатка используется особенность алгоритма RSA, использующего операцию **возведения в степень только совместно с получением остатка от деления** результата на заданное число.

(a^k)mod(n):

r=1

?(k>0)

?(k-нечетное) r=(r\*a)mod(n)

a=(a\*a)mod(n)

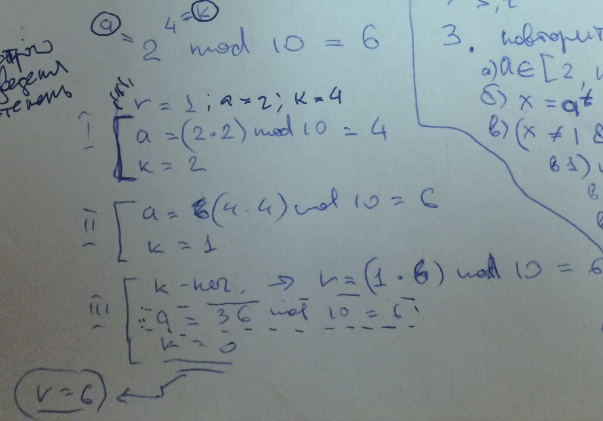
k>>=1



Доказательство возможности использования модуля

(а\*b)mod(n)=[(а)mod(n)\*(b)mod(n)]mod(n)=

Пример:



Шифрование с открытым ключом позволило сгенерировать и передать секретный ключ для симметричного шифрования.

<<схема отправитель->получатель+генератор=пароль->отправитель>>

