Лекция 5

Криптография с открытым ключом

Вопросы:

1. Принципы построения криптосистем с открытым ключом.
2. Математические основы.
3. Криптосистема RSA.
4. Вычислительные аспекты и защищенность RSA.

Вопрос 1

Принципы построения криптосистем с открытым ключом


Криптосистемы
Симметричные криптосистемы
Асимметричные криптосистемы
Операции подстановки и перестановки
Специальные математические функции

Криптография с открытым ключом

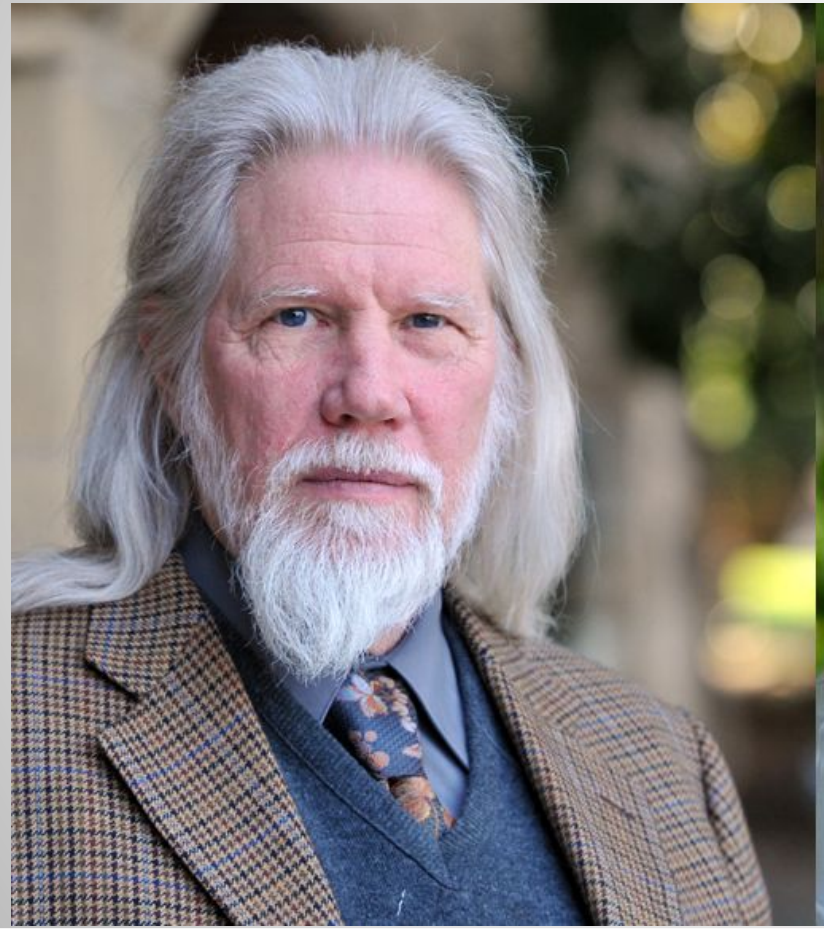
Криптография с секретным ключом

Заблуждения:

1) Шифрование с открытым ключом более защищено от криптоанализа, чем традиционное шифрование;

2) Криптосистемы с открытым ключом являются универсальным подходом, делающим традиционное шифрование безнадежно устаревшим.

Уитфилд Диффи Мартин Хеллман



Модель криптосистемы с открытым ключом

1. Алгоритмы шифрования и расшифрования являются открытыми.

2. Каждый участник генерирует пару ключей.

3. Один ключ из пары называют **открытым ключом.** Он публикуется (размещается в открытом каталоге или файле).

4. Второй ключ из пары называют **личным ключом**, т.к. он остается в личном владении (генерируются участником для себя и никогда никуда не передается).

5. Один ключ из пары используется для шифрования, другой – для расшифрования.

6. С точки зрения вычислений нереально определить ключ расшифрования, зная только используемый криптографический алгоритм и ключ шифрования.

7. До тех пор, пока системе удается сохранять свой личный ключ в секрете, сообщения оказываются защищенными.

8. В любой момент система может сгенерировать новую пару ключей и опубликовать свой новый открытый ключ.

Отправитель - Алиса - создает открытый текст X:

$$X = [x_1, x_2, \ldots, x_M],\ x_i \in A \text{ (конечный алфавит).}$$

Сообщение адресовано **получателю - Бобу**. Получатель генерирует пару ключей:

KU_b - открытый ключ

KR_b - личный ключ

Отправитель формирует шифрованный текст Y:

$$Y = [y_1, y_2, \ldots, y_N]$$

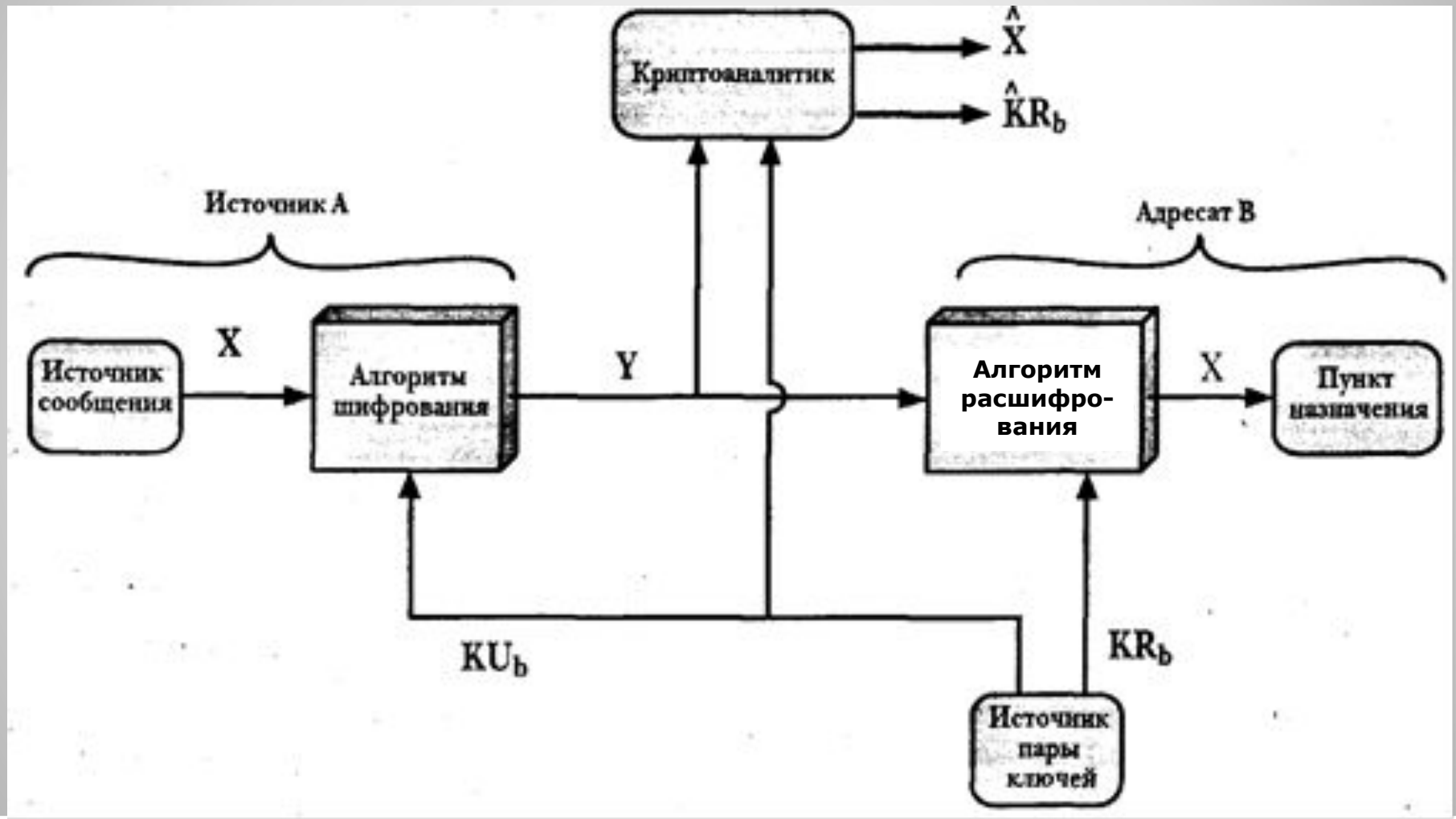
$$Y = E_{KUb}(X)$$

Получатель выполняет обратное преобразование:

$$X = D_{KRb}(Y)$$

Противник, наблюдая Y и имея доступ к KU_b, но не к KR_b, или X, должен будет пытаться восстановить X и/или KR_b.

Схема обеспечения конфиденциальности



Для обеспечения аутентификации:

Пара ключей генерируется на стороне отправителя. В данном случае все шифрованное сообщение выступает в качестве *цифровой подписи:*

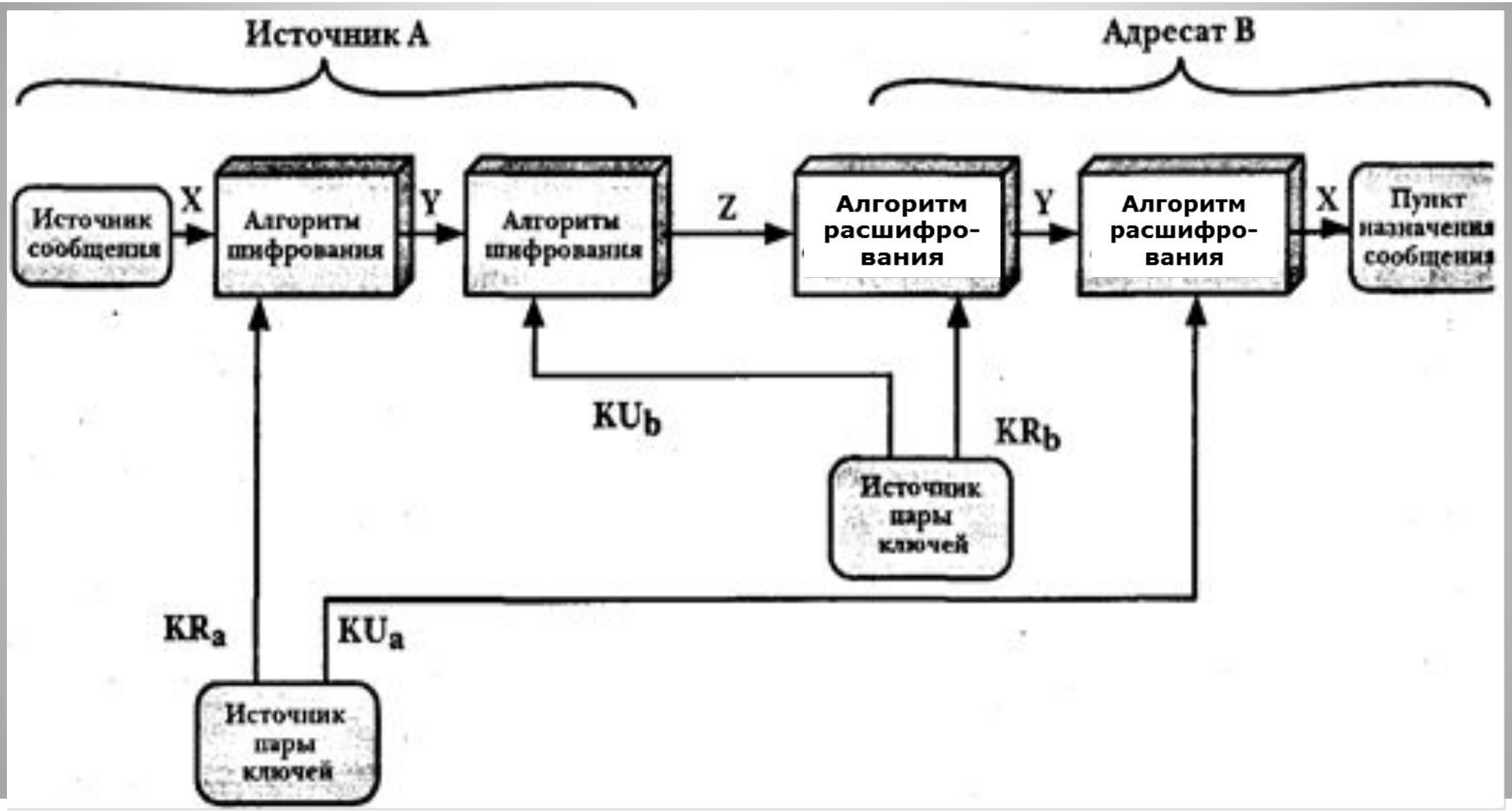
$$Y=E_{KRa}(X),$$

$$X=D_{KUa}(Y).$$

Для обеспечения конфиденциальности и аутентификации необходимо двойное шифрование:

$$Y=E_{KUb}[E_{KRa}(X)],$$

$$X = E_{KUa}[E_{KRb}(Y)].$$


Источник A
Адресат B
Источник сообщения
X
Алгоритм шифрования
Y
Алгоритм шифрования
Z
Алгоритм расшифро-вания
Y
Алгоритм расшифро-вания
X
Пункт назначения сообщения
KUb
KRb
Источник пары ключей
KRa
KUa
Источник пары ключей

Применение криптосистем с открытым ключом:

1) Шифрование/расшифрование. Отправитель шифрует сообщение с использованием открытого ключа получателя.

2) Цифровая подпись. Отправитель "подписывает" сообщение с помощью своего личного ключа.

3) Обмен ключами (Управление ключами). Две стороны взаимодействуют, чтобы обменяться сеансовым ключом.

Примеры: RSA, DSS, алгоритм Диффи-Хеллмана.

Недостатки асимметричных криптосистем:

1) Криптосистемы с открытым ключом медленные.

2) Криптосистемы с открытым ключом уязвимы к атакам на основе подобранного открытого текста.

Если известен шифротекст Y=E(X), где X – открытый текст, и **существует n возможных открытых текстов**, криптоаналитику достаточно зашифровать все n возможных открытых текстов открытым ключом и сравнить результаты с Y.

Он не сможет таким путем восстановить ключ расшифрования, но сумеет определить X.

Вопрос 2

Математические основы

Односторонней функцией называется функция, отображающая свои аргументы в некоторый диапазон значений так, что каждое значение функции имеет уникальное обратное значение, при этом значения функции вычислить легко, а обратные — практически невозможно.

$Y = f(X)$ - вычисляется легко,

$X = f^{-1}(Y)$ - практически не поддается вычислению.

Основным критерием отнесения функции f к классу однонаправленных функций является отсутствие эффективных алгоритмов обратного преобразования Y в X:

- таких алгоритмов не существует,
- алгоритмы пока не найдены,
- вычисления займут миллионы лет.

Примеры однонаправленных функций

1) Целочисленное умножение.

Вычисление произведения двух очень больших целых чисел P и Q:

N = P*Q

Обратная задача – разложение на множители большого целого числа N, т.е. нахождение делителей P и Q, является практически неразрешимой при достаточно больших значениях N.

По современным оценкам при $N=2^{664}$ и $P=Q$ для разложения числа N потребуется около 10^{23} операций. (10^{10} - возраст Вселенной, 10^{51} – число атомов планеты, 10^{57} – число атомов Солнца)

2) Модульная экспонента с фиксированным основанием и модулем.

Пусть A и N – целые числа и $1 \le A < N$.

Модульная экспонента с основанием A по модулю N представляет собой функцию

$$f_{A,N}(x) = A^x (mod\ N),$$ где x – целое число, $1 \le x \le N-1$.

Обратная задача – задача дискретного логарифмирования - формулируется так: для известных целых A, N, Y найти целое число x, такое, что:

$$A^x (mod\ N) = Y$$

Модульная экспонента отнесена к однонаправленным функциям **условно** (эффективные алгоритмы не найдены, но не доказано, что они не существуют).

По современным оценкам теории чисел при $A=2^{664}$ и $N=2^{664}$ решение задачи потребует около 10^{26} операций.

3) Однонаправленные функции с «потайным ходом»(с лазейкой).

Функция $f: X \rightarrow Y$ относится к классу однонаправленных функций с «потайным ходом» в том случае, если она является однонаправленной и возможно эффективное вычисление обратной функции, когда известен «потайной ход» (секретное число, строка или другая информация).

Пример: используемая в RSA модульная экспонента с фиксированным модулем и показателем степени.

Модулярная арифметика или ***арифметика в классах вычетов***.

Если целые числа a и b имеют одинаковые остатки при делении на n, то они называются **сравнимыми по модулю n:**

$$a \equiv b \pmod{n}$$

Вычетом числа a по модулю n называется остаток от деления a на n.

В общем случае, $a \equiv b \pmod{n}$, если $a = kn+b$ при некотором целом k. Если $a \geq 0$, а b находится между 0 и n, можно рассматривать b как остаток от деления a на n.

Операция $a \bmod n$ называется приведением a по модулю n.

Во всех криптографических алгоритмах, если операция mod возвращает отрицательное число, необходимо прибавить *n* к результату операции.

Пусть *a* – целое число больше 1. *a* является ***простым числом***, если его единственными положительными делителями будут 1 и само *a.*

Наибольший общий делитель чисел *a* и *b*, обозначаемый как **НОД(*a*,*b*)** – это наибольшее целое, делящее одновременно числа *a* и *b*.

Если *НОД*(*a*,*b*)=1, то целые числа *a* и *b* – ***взаимно простые***. Наибольший общий делитель может быть вычислен с помощью ***алгоритма Евклида***.

Если ab≡1mod n, то b называется ***мультипликативным обратным a по модулю n:***

$$a \equiv b^{-1} \bmod n$$

Вычисляется мультипликативное обратное с помощью **расширенного алгоритма Евклида**.

Функция Эйлера *φ(n)* – это количество положительных целых, меньших n, которые взаимно просты с *n*.

Вопрос 3

Криптосистема RSA

RSA - алгоритм шифрования с открытым ключом.

Опубликован 1978 году.

Разработчики: Рон Райвест (Rivest), Ади Шамир (Shamir), Лен Адлеман (Adleman).

RSA – универсальный алгоритм, может использоваться для шифрования, ЭЦП, распределения ключей.

Применяется во многих криптографических приложениях: PGP, S/MIME, TLS/SSL, IPSec и др.

RSA – блочный шифр, в котором открытый текст M, шифртекст C, открытый ключ KUb и личный ключ KRb принадлежат множеству целых чисел $Z_N = \{0,1,\ldots,N-1\}$, где N – модуль:

$$\mathbf{N = P*Q}$$

P и Q – случайные большие простые числа. Для обеспечения максимальной безопасности выбирают P и Q примерно равной длины.

Открытый текст M шифруется блоками M_i длиной k бит:

$\mathbf{0 \le M_i \le N-1}$ (двоичное значение каждого блока <N)

На практике длина блока в битах - k выбирается из расчёта: $\boldsymbol{2^k < N \le 2^{k+1}}$

Отправитель должен знать *N* и *e*, т.е.

***KU*={*e*,*N*} – открытый ключ**

Получатель должен знать *P*,*Q* и *d*:

***KR*={*d*,*P*,*Q*} – личный ключ**

Пусть пользователь А хочет передать пользователю В сообщение в зашифрованном виде с использованием RSA (конфиденциальность).

Криптосистему RSA должен сформировать получатель сообщения, т.е. пользователь В.

Описание алгоритма RSA

На стороне пользователя B:

1. Пользователь B выбирает 2 произвольных больших простых числа *P* и *Q*.

2. Пользователь B вычисляет значение модуля N

$$N = P*Q$$

3. Пользователь B вычисляет функцию Эйлера (количество положительных целых <N, которые взаимно просты с N)

$$\varphi(N) = (P-1)(Q-1)$$

4. Пользователь B выбирает случайным образом значение элемента открытого ключа *e* с учётом выполнения условий:

$$НОД(\phi(n), e) = 1$$ (*e* и *φ(N)* – взаимно простые)

$$1 < e \leq \phi(n)$$

.

Описание алгоритма RSA (продолжение)

5. Пользователь B вычисляет значение элемента секретного ключа *d*, используя расширенный алгоритм Евклида при решении сравнения

$$d \equiv e^{-1} \bmod \phi(N)$$

(e и d являются взаимно простыми по модулю φ(N))

или

$$ed \equiv 1 \bmod \phi(N),\ d < \phi(N)$$

6. Пользователь B пересылает пользователю A открытый ключ - пару чисел $KU_b=(N,e)$ по незащищённому каналу.

Описание алгоритма RSA (продолжение)

На стороне пользователя А:

7. Пользователь А разбивает исходный открытый текст М на блоки, каждый из которых может быть представлен в виде числа

$$0 \le M_i \le N-1$$

8. Пользователь А шифрует текст, представленный в виде последовательности чисел M_i по формуле

$$C_i = M_i^e (\mathrm{mod}\ N)$$

и отправляет криптограмму С пользователю В

$$C_1, C_2, \ldots, C_i, \ldots$$

На стороне пользователя В:

9. Пользователь В расшифровывает принятую криптограмму, используя личный ключ по формуле

$$M_i = C_i^d (\mathrm{mod}\ N)$$

Пример

Пользователь B

1. Выбирает $P=3$ и $Q=11$
2. Вычисляет модуль $N=P\cdot Q=3\cdot 11=33$
3. Вычисляет значение функции Эйлера

$$\varphi(N)=\varphi(33)=(P-1)(Q-1)=2\cdot 10=20$$

4. Выбирает в качестве открытого ключа e - произвольное число взаимно простое с $\varphi(N)$ и $1 < e \leq \phi(n)$.

$$1<e\leq 20,\ НОД(e,20)=1$$

Пусть $e=7$

Открытый ключ $KU_b=\{7, 33\}$

Пример

5. В вычисляет значение секретного ключа *d*, используя расширенный алгоритм Евклида при решении сравнения

$$ed \equiv 1 \bmod \varphi(N)$$

это значит $ed = k*\varphi(N)+1$

(если $a \equiv b \bmod n$, то $a = k*n+b$)

Возьмем k=1, подставим e и φ(N), получим

$7*d = 1*20 + 1 = 21$, отсюда ***d=3***

3<20, т.е. условие d< φ(*N*) выполняется.

Личный ключ $KR_b = \{3,3,11\}$

6. В пересылает пользователю А открытый ключ - пару чисел (*N*=33, *e*=7)

Пример

Пользователь А

7. Представляет шифруемое сообщение как последовательных чисел в диапазоне 0..32 (n-1=32)

Пусть шифруется сообщение CAB

Пусть A-1, B-2, C-3

CAB: $M_1=3$, $M_2=1$, $M_3=2$

8. А шифрует текст $M_1M_2M_3$, используя e=7 и N=33, по формуле

$$C_i=M_i^e(mod\ N)=M_i^7(mod\ 33)$$

Получает $C_1=3^7\ mod\ 33=2187\ mod\ 33=9$

$C_2=1^7\ mod\ 33=1$

$C_3=2^7\ mod\ 33=128\ mod\ 33=29$

А отправляет В криптограмму C_1,C_2,C_3=9,1,29.

Пример

Пользователь B

9. Расшифровывает криптограмму, используя d=3 по формуле $Mi=Ci^3 \ mod 33$

$M_1 = 9^3 \ mod \ 33 = 729 \ mod \ 33 = 3$
$M_2 = 1^3 \ mod \ 33 = 1$
$M_3 = 29^3 \ mod \ 33 = 24389 \ mod \ 33 = 2$

Таким образом исходное сообщение: 3,1,2=CAB

Вопрос 4

Вычислительные аспекты и защищенность RSA

Вычислительные аспекты

I) При вычислении ключей необходимо выбрать 2 больших ($10^{75} \div 10^{100}$) простых числа *P* и *Q*.

Этапы:

a) выбор случайного нечётного числа приблизительно желаемой величины;

b) выяснение, является ли это число простым.

Для проверки числа на простоту существует целый ряд тестов (Миллера-Рабина, Соловея-Штрассена, Лемана).

Почти все такие тесты носят вероятностный характер. Это значит, что тест определит только, что число вероятно простое.

c) затем подбирается 2-ое простое число из условий:

- *P* и *Q* должны различаться по длине всего на несколько разрядов;
- *НОД*(*P*-1,*Q*-1) должен быть достаточно малым.

Вычислительные аспекты

II) После определения P и Q выбирается значение открытого ключа. Процедура заключается в генерировании случайных чисел и сравнении с $\varphi(N)$, пока не будет найдено число, взаимно простое с $\varphi(N)$:

$$НОД(e, \varphi(N))=1$$

III) Шифрование и расшифрование в RSA предполагает возведение целого числа в целую степень по $mod\ n$.

Если возведение в степень выполнять непосредственно с целыми числами, а потом проводить сравнение по модулю n, то промежуточные значения окажутся огромными.

Поэтому, используя свойство арифметики в классах вычетов:

$$((a\ mod\ n)\cdot(b\ mod\ n))\ mod\ n=(a\cdot b)\ mod\ n$$

можно приводить промежуточные результаты по $mod\ n$. Это делает вычисления практически возможными.

Вычислительные аспекты

Вторая проблема – эффективная реализация операции возведения в степень.

В RSA используются очень большие показатели степени. Один из возможных подходов - возведение числа в квадрат и повторное возведение в квадрат промежуточных результатов:

$$x^{16}=x\cdot xxxxxxxxxxxxxxx=(((x^2)^2)^2)^2.$$

Существуют специальные методы ускорения вычислений степени числа по модулю другого числа $a^x \bmod n$:

1) Метод двоичных квадратов и умножений (аддитивная цепочка);

2) Метод Барретта;

3) Метод Монтгомери.

Защищённость алгоритма RSA

Тремя возможными подходами к криптоанализу алгоритма RSA являются следующие.

1) **Простой перебор.** Предполагает проверку всех возможных личных ключей.

2) **Математический анализ.** Существует несколько подходов такого рода, но все они по сути эквивалентны нахождению множителей p и q, $p \cdot q = n$.

3) **Анализ временных затрат.** Опирается на анализ времени выполнения алгоритма расшифрования.

Если удастся разложить n *на* множители p и q, *это* позволит вычислить $\varphi(n)=(p-1)(q-1)$ и затем определить личный ключ $d=e^{-1} \ mod \ \varphi(n)$.

<u>С точки зрения матанализа существуют 2 угрозы:</u>

1) непрерывный рост вычислительной мощности современных компьютеров.

2) непрерывное усовершенствование алгоритмов разложения на множители.

Против анализа временных затрат имеются следующие контрмеры:

- **Постоянное время выполнения операции возведения в степень.** Изменение алгоритма таким образом, чтобы все возведения в степень занимали одно и то же время. При этом увеличивается общее время выполнения алгоритма.
- **Случайные задержки.** Добавление в алгоритм возведения в степень случайных задержек, что уменьшает пользу от анализа временных затрат.
- **Маскировка.** Умножение шифрованного текста на случайное число перед тем, как выполнять возведение в степень.