1. Начинаем с трудной задачи P. Она должна решаться сложно в смысле теории: не должно быть алгоритма, с помощью которого можно было бы перебрать все варианты решения задачи Pза [полиномиальное время](http://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D0%BB%D0%B8%D0%BD%D0%BE%D0%BC%D0%B8%D0%B0%D0%BB%D1%8C%D0%BD%D1%8B%D0%B9_%D0%B0%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC) относительно размера задачи. Более правильно сказать: не должно быть *известного* полиномиального алгоритма, решающего данную задачу — так как ни для одной задачи ещё пока не доказано, что для неё подходящего алгоритма нет в принципе.
2. Можно выделить легкую подзадачу P'из P. Она должна решаться за полиномиальное время и лучше, если за линейное.
3. «Перетасовываем и взбалтываем» P', чтобы получить задачу P'', совершенно не похожую на первоначальную. Задача P''должна по крайней мере выглядеть как оригинальная труднорешаемая задача P.
4. P''открывается с описанием, как она может быть использована в роли ключа зашифрования. Как из P''получить P', держится в секрете как секретная лазейка.
5. Криптосистема организована так, что алгоритмы расшифрования для легального пользователя и криптоаналитика существенно различны. В то время как второй решает P''-задачу, первый использует секретную лазейку и решает P'-задачу

Пусть K — пространство ключей, а eи d — ключи шифрования и расшифрования соответственно. E_e — функция шифрования для произвольного ключа e\in*K, такая что:*

E_e(m)=c

Здесь c\inC, где C — пространство шифротекстов, а m\inM, где M — пространство сообщений.

D_d — функция расшифрования, с помощью которой можно найти исходное сообщение m, зная шифротекст c :

D_d(c)=m

{E_e: e\in*K*} — набор шифрования, а {D_d: d\in*K*} — соответствующий набор для расшифрования. Каждая пара (E, D)имеет свойство: зная E_e, невозможно решить уравнение E_e(m)=c, то есть для данного произвольного шифротекста c\inC, невозможно найти сообщение m\inM. Это значит, что по данному eневозможно определить соответствующий ключ расшифрования d. E_eявляется односторонней функцией, а d — лазейкой

В этой модели Ева перехватывает открытый ключ e, посланный Бобом Алисе. Затем создает пару ключей e'и d', «маскируется» под Боба, посылая Алисе открытый ключ e', который, как думает Алиса, открытый ключ, посланный ей Бобом. Ева перехватывает зашифрованные сообщения от Алисы к Бобу, расшифровывает их с помощью секретного ключа d', заново зашифровывает открытым ключом eБоба и отправляет сообщение Бобу. Таким образом, никто из участников не догадывается, что есть третье лицо, которое может как просто перехватить сообщение m, так и подменить его на ложное сообщение m'.

Ещё одна форма атаки — вычисление закрытого ключа, зная открытый (рисунок ниже). Криптоаналитик знает алгоритм шифрования E_e, анализируя его, пытается найти D_d. Этот процесс упрощается, если криптоаналитик перехватил несколько криптотекстов с, посланных лицом A лицу B.

Большинство криптосистем с открытым ключом основаны на проблеме факторизации больших чисел. К примеру, [RSA](http://ru.wikipedia.org/wiki/RSA)

Рассмотрим алгоритм RSA с практической точки зрения.

Для начала необходимо сгенерировать открытый и секретные ключи:

* Возьмем два больших простых числа p and q.
* Определим n, как результат умножения p on q (n= p\*q).
* Выберем случайное число, которое назовем d. Это число должно быть взаимно простым (не иметь ни одного общего делителя, кроме 1) с результатом умножения (p-1)\*(q-1).
* Определим такое число е, для которого является истинным следующее соотношение (e\*d) mod ((p-1)\*(q-1))=1.
* Hазовем открытым ключем числа e и n, а секретным - d и n.

Для того, чтобы зашифровать данные по открытому ключу {e,n}, необходимо следующее:

* разбить шифруемый текст на блоки, каждый из которых может быть представлен в виде числа M(i)=0,1,2..., n-1( т.е. только до n-1).
* зашифровать текст, рассматриваемый как последовательность чисел M(i) по формуле C(i)=(M(I)^e)mod n.

Чтобы расшифровать эти данные, используя секретный ключ {d,n}, необходимо выполнить следующие вычисления: M(i) = (C(i)^d) mod n. В результате будет получено множество чисел M(i), которые представляют собой исходный текст.

Следующий пример наглядно демонстрирует алгоритм шифрования RSA:

Зашифруем и расшифруем сообщение "САВ" по алгоритму RSA. Для простоты возьмем небольшие числа - это сократит наши расчеты.

* Выберем p=3 and q=11.
* Определим n= 3\*11=33.
* Hайдем (p-1)\*(q-1)=20. Следовательно, d будет равно, например, 3: (d=3).
* Выберем число е по следующей формуле: (e\*3) mod 20=1. Значит е будет равно, например, 7: (e=7).
* Представим шифруемое сообщение как последовательность чисел в диапозоне от 0 до 32 (незабывайте, что кончается на n-1). Буква А =1, В=2, С=3.

Теперь зашифруем сообщение, используя открытый ключ {7,33}

C1 = (3^7) mod 33 = 2187 mod 33 = 9;   
C2 = (1^7) mod 33 = 1 mod 33 = 1;   
C3 = (2^7) mod 33 = 128 mod 33 = 29; 

Теперь расшифруем данные, используя закрытый ключ {3,33}.

M1=(9^3) mod 33 =729 mod 33 = 3(С);   
M2=(1^3) mod 33 =1 mod 33 = 1(А);   
M3=(29^3) mod 33 = 24389 mod 33 = 2(В);   
  
Данные расшифрованы!

**Недостатки**

Недостатки алгоритма несимметричного шифрования в сравнении с симметричным:

* В алгоритм сложнее внести изменения.
* Хотя сообщения надежно шифруются, но получатель и отправитель самим фактом пересылки шифрованного сообщения «засвечиваются».[[8]](http://ru.wikipedia.org/wiki/%D0%9A%D1%80%D0%B8%D0%BF%D1%82%D0%BE%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0_%D1%81_%D0%BE%D1%82%D0%BA%D1%80%D1%8B%D1%82%D1%8B%D0%BC_%D0%BA%D0%BB%D1%8E%D1%87%D0%BE%D0%BC#cite_note-7)[[*уточнить*](http://ru.wikipedia.org/wiki/%D0%92%D0%B8%D0%BA%D0%B8%D0%BF%D0%B5%D0%B4%D0%B8%D1%8F:%D0%90%D0%B2%D1%82%D0%BE%D1%80%D0%B8%D1%82%D0%B5%D1%82%D0%BD%D1%8B%D0%B5_%D0%B8%D1%81%D1%82%D0%BE%D1%87%D0%BD%D0%B8%D0%BA%D0%B8)]
* Более длинные ключи. Ниже приведена таблица, сопоставляющая длину ключа симметричного алгоритма с длиной ключа RSA с аналогичной криптостойкостью:

|  |  |
| --- | --- |
| **Длина симметричного ключа, бит** | **Длина ключа RSA, бит** |
| 56 | 384 |
| 64 | 512 |
| 80 | 768 |
| 112 | 1792 |
| 128 | 2304 |

* Шифрование-расшифрование с использованием пары ключей проходит на два-три порядка медленнее, чем шифрование-расшифрование того же текста симметричным алгоритмом.
* Требуются существенно бо́льшие вычислительные ресурсы, поэтому на практике асимметричные криптосистемы используются в сочетании с другими алгоритмами:
  1. Для [ЭЦП](http://ru.wikipedia.org/wiki/%D0%AD%D0%BB%D0%B5%D0%BA%D1%82%D1%80%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D1%86%D0%B8%D1%84%D1%80%D0%BE%D0%B2%D0%B0%D1%8F_%D0%BF%D0%BE%D0%B4%D0%BF%D0%B8%D1%81%D1%8C) сообщение предварительно подвергается хешированию, а с помощью асимметричного ключа подписывается лишь относительно небольшой результат [хеш-функции](http://ru.wikipedia.org/wiki/%D0%A5%D0%B5%D1%88-%D1%84%D1%83%D0%BD%D0%BA%D1%86%D0%B8%D1%8F).
  2. Для шифрования они используются в форме [гибридных криптосистем](http://ru.wikipedia.org/wiki/%D0%93%D0%B8%D0%B1%D1%80%D0%B8%D0%B4%D0%BD%D0%B0%D1%8F_%D0%BA%D1%80%D0%B8%D0%BF%D1%82%D0%BE%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0), где большие объёмы данных шифруются [симметричным шифром](http://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D0%BC%D0%BC%D0%B5%D1%82%D1%80%D0%B8%D1%87%D0%BD%D1%8B%D0%B5_%D0%BA%D1%80%D0%B8%D0%BF%D1%82%D0%BE%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D1%8B) на сеансовом ключе, а с помощью асимметричного шифра передаётся только сам сеансовый ключ.

Пусть есть 3 ключа K_A, K_B, K_C, распределенные так, как показано в таблице.

|  |  |
| --- | --- |
| **Лицо** | **Ключ** |
| Алиса | K_A |
| Боб | K_B |
| Кэрол | K_C |
| Дэйв | K_A, K_B |
| Эллен | K_B, K_C |
| Франк | K_A, K_C |

Тогда Алиса может зашифровать сообщение ключом K_A, а Эллен расшифровать ключами K_B, K_C, Кэрол — зашифровать ключом K_C, а Дэйв расшифровать ключами K_A, K_B. Если Дэйв зашифрует сообщение ключом K_A, то сообщение сможет прочитать Эллен, если ключом K_B, то его сможет прочитать Франк, если же обоими ключами K_Aи K_B, то сообщение прочитает Кэрол. По аналогии действуют и другие участники. Таким образом, если используется одно подмножество ключей для шифрования, то для расшифрования требуются оставшиеся ключи множества. Такую схему можно использовать для n ключей.

|  |  |
| --- | --- |
| **Шифруется ключом** | **Расшифровывается ключом** |
| K_Bи K_C | K_A |
| K_Aи K_C | K_B |
| K_Aи K_B | K_C |
| K_C | K_A, K_B |
| K_A | K_B, K_C |
| K_B | K_A, K_C |

* Теперь можно посылать сообщения группам агентов, не зная заранее состав группы.

Рассмотрим для начала множество, состоящее из трех агентов: Алисы, Боба и Кэрол. Алисе выдаются ключи K_Aи K_B, Бобу — K_Bи K_C, Кэрол — K_Aи K_C. Теперь, если отправляемое сообщение зашифровано ключом K_C, то его сможет прочитать только Алиса, последовательно применяя ключи K_Aи K_B. Если нужно отправить сообщение Бобу, сообщение шифруется ключом K_A, Кэрол — ключом K_B. Если нужно отправить сообщение и Алисе и Кэрол, то для шифрования используются ключи K_Bи K_C.

Преимущество этой схемы заключается в том, что для её реализации нужно только одно сообщение и n ключей (в схеме с n агентами). Если передаются индивидуальные сообщения, то есть используются отдельные ключи для каждого агента (всего n ключей) и каждого сообщения, то для передачи сообщений всем различным подмножествам требуется 2^n-2ключей.

Недостатком такой схемы является то, что необходимо также широковещательно передавать подмножество агентов (список имён может быть внушительным), которым нужно передать сообщение. Иначе каждому из них придется перебирать все комбинации ключей в поисках подходящей. Также агентам придется хранить немалый объём информации о ключах

Размер ключа в алгоритме RSA связан с размером модуля n. Два числа p и q, произведением которых является модуль, должны иметь приблизительно одинаковую длину поскольку в этом случае найти сомножители (факторы) сложнее, чем в случае когда длина чисел значительно различается. Например, если предполагается использовать 768-битный модуль, то каждое число должно иметь длину приблизительно 384 бита. Обратите внимание, что если два числа чрезвычайно близки друг к другу или их разность близка к некоторому предопределенному значению, то возникает потенциальная угроза безопасности, однако такая вероятность – близость двух случайно выбранных чисел – незначительна.