# RSA

**Шифрование.** Поговорим о системе шифрования RSA, которая сплошь и рядом используется в современном мире.

Пусть у нас есть информация в виде целого числа $x$. Почему это число, а не текст? С одной стороны, все слышали, что любая информация в компьютере, в конечном счёте, двоичное число. С другой стороны, если не ясно, где в этом компьютере искать нужное нам число, можно и руками любой текст превратить в последовательность цифр, по которой легко будет восстановить исходный текст. Можно, например, вместо каждой буквы писать её номер в алфавите (вместо "а" — "01", вместо "к" — "11"), а вместо пробела — "34", вместо знаков препинания тоже соответствующие номера.

Итак, пусть мы хотим зашифровать число $x$. Выберем некоторое $n$ и два числа $e$ и $d$, такие что $\boldsymbol{ed-1}$ **делится на** $\boldsymbol{\varphi(n)}$. Для этого рассмотрим произвольное $e$, взаимно простое с $\varphi(n)$, а для него будут существовать натуральные $d, k$, такие что $ed + \varphi(n)k = 1$.

Предлагается **зашифровать** $\boldsymbol{x}$ **числом** $\boldsymbol{y = x^e \ mod \ n}$ ($y$ — остаток от деления $x^e$ на $n$). Зная $d$, можно **по** $\boldsymbol{y}$ **восстановить исходную информацию** $\boldsymbol{x}$**:** $\boldsymbol{x = y^d \ mod \ n}$. Действительно, $y^d \equiv x^{ed} \equiv x^{ed-1} \cdot x \equiv x (mod\ n)$). Число $d$ хранится в секрете и называется закрытым ключом. Числа $n$ и $e$ свободно распространяются и называются открытым ключом. Таким образом, кто угодно может зашифровать сообщение, но лишь мы (или другой владелец закрытого ключа) сможем его дешифровать.

Обсудим, как выбрать $n$, чтобы шифрование было надёжным. Для этого возьмём $n$ равное произведению двух достаточно больших простых чисел. Почему в этом случае по известным $n$ и $e$ совсем непросто вычислить $d$? (а ведь именно это нужно, чтоб противник не раскрыл ваши тайны!) Дело в том, что для этого потребовалось бы вычислить $\varphi(n)$ (ведь $d$ можно найти только из условия $ed - 1 \vdots \varphi(n)$), то есть разложить $n$ на простые множители. Алгоритмов для этого в данный момент не существует. Но, скажете вы, а почему нельзя просто перебором на компьютере?

На среднем компьютере в секунду выполняется примерно $10^8$ операций, на очень мощном примерно $10^{12}$. Предположим, наши простые числа имеют около 20 знаков, а значит $n$ имеет около 40 знаков. Чтобы гарантированно найти делитель числа $n$, нужно перебрать все числа от 1 до $[\sqrt{n}\,]$, то есть проделать порядка $10^{20}$ операций. Это займёт даже на самом мощном компьютере $10^8$ секунд $= 1157$ дней $> 3$ лет. Заметим, что в жизни применяются простые числа длиной 100–200 знаков, поэтому для взламывания реальной криптосистемы одному компьютеру понадобится порядка $10^{80}$ лет — примерно столько, сколько атомов во вселенной.

| Этап | Описание операции | Результат операции |
|---|---|---|
| Генерация ключей | выбрать два простых числа | $p = 3557, q = 2579$ |
| | вычислить модуль | $n = p \cdot q = 3557 \cdot 2579 = 9173503$ |
| | вычислить функцию Эйлера | $\varphi(n) = (p-1)(q-1) = 9167368$ |
| | выбрать открытый показатель | $e = 3$ |
| | вычислить секретный показатель | $d = 6111579$ |
| | опубликовать открытый ключ | $(e, n) = (3, 9173503)$ |
| | сохранить секретный ключ | $(d, n) = (6111579, 9173503)$ |
| Шифрование | выбрать открытый текст | $x = 111111$ |
| | вычислить шифротекст | $y = x^e \ mod\ n = 111111^3 \ mod\ 9173503 = 4051753$ |
| Дешифровка | вычислить исходное сообщение | $x = y^d \ mod\ n = 4051753^{6111579} \ mod\ 9173503 = 111111$ |

**Цифровая подпись.** Что есть подпись? В идеале это должен быть атрибут документа, обладающий следующими свойствами:

1) Достоверность. Наличие подписи убеждает, что лицо, её сделавшее, видело документ и сознательно с ним согласилось.

2) Неподделываемость. Подпись может нанести только титульное лицо.

3) Неотделимость от документа. Подпись нельзя "срезать" с документа и повторно использовать, "сопроводив" ею другой документ (без согласия титульного лица).

4) Окончательность. Невозможно изменить подписанный документ после подписывания.

5) Неотрицаемость. Лицо, поставившее подпись, не должно иметь возможности сделать вид, что оно этого не делало.

Довольно много требований, да? Поразительно, но оказывается, RSA с лёгкостью решает все эти задачи.

| Алгоритм Алисы | Алгоритм Боба |
|---|---|
| Взять открытый текст $x$ | принять пару $(x, s)$ |
| Создать цифровую подпись $s$ с помощь своего секретного ключа $(d, n)$: $s = x^d \ mod \ n$ | взять открытый ключ $(e, n)$ |
| | Проверить подлинность подписи: $s^e \equiv? \equiv x(mod\ n)$ |

Не правда ли, мы получили неподделываемость? Предположим, Кэрролл хочет подделать подпись Алисы. Что ему потребуется? Найти такое число $s_1$, что $s_1^e \equiv x(mod\, n)$ (при том, что числа $x, e, n$ он знает). Много ли таких чисел? Например, если $n = 8, x = 1, e = 2$? Получается, 4 решения: $1, 3, 5, 7$. Значит, Кэрролл, просто перебирая различные значения для $s_1$, может вскоре наткнуться на нужное ему? Оказывается, нет. Вспомним, что $ed - 1 \vdots \varphi(n)$, значит, если $s_1^e \equiv x(mod\, n)$, то $s_1 \equiv s_1 \cdot s_1^{ed-1} \equiv s_1^{ed} \equiv x^d(mod\, n)$, где $x$ и $d$ известные числа. То есть решение уравнения, интересующего Кэрролла, единственно, и просто перебором он вряд ли чего-нибудь добьётся за разумное время. Таким образом, с неподделываемостью всё в порядке.

Но некоторый минус всё же присутствует. Алиса может пожертвовать свой (дорогостоящий!) секретный ключ, чтобы получить возможность отрицать свои подписи (например, выложив его где-нибудь в интернете таким образом, будто это не она сделала). То есть вопрос неотрицаемости всё же не решён.

Всем, кого заинтересовала тема криптографии, можно посоветовать почитать (хотя бы отрывками) книгу Брюса Шнайера "Прикладная криптография" (например, http://lib.zhilinsky.ru/books/krypto/prikladnaya-kriptografiya ). Никакой сложной математики там заведомо нет. Существует также художественная книга, затрагивающая вопросы криптографии. Это "Криптонамикон", написаный бывшим хакером Нилом Стивенсоном.