ФАКТОРИЗАЦИЯ БОЛЬШИХ ЦЕЛЫХ ЧИСЕЛ И КРИПТОГРАФИЯ*

И. В. Агафонова
ivagafonova@home.eltel.net

12 декабря 2006 г.

1. Односторонние функции

В криптографии «с открытым ключом» (асимметрические криптосистемы) центральными являются понятия:

- односторонняя функция (однонаправленная);
- односторонняя функция с секретом.

Функция $f\colon X \to Y$ ***односторонняя***, если существует эффективный алгоритм для её вычисления при любом X, но не существует эффективного алгоритма для вычисления обратной к ней функции. (Под эффективным алгоритмом понимается полиномиальный, в отличие от экспоненциального.) Функция называется ***односторонней с секретом*** (потайным ходом, лазейкой, trapdoor), если при наличии некоторой дополнительной информации (ключа) возможно её эффективное обращение. Односторонними (на современном уровне наших знаний) являются функции:

1) Умножение целых чисел
$$f(x, y) = x\,y$$

2) Возведение заданного числа в степень по данному модулю
$$f_{n,a}(m) = \langle a^m \rangle_n$$
(Эта запись означает вычет a^m по модулю n)

*Семинар по дискретному гармоническому анализу и геометрическому моделированию «DHA & CAGD»: `http://www.dha.spb.ru/`

3) Возведение числа в заданную степень по данному модулю

$$f_{n,m}(a) = \langle a^m \rangle_n$$

Нахождение функций, обратных к данным, приводит к задачам:

1) Факторизация целого числа: $f = x\,y$, найти x и y, зная f.

2) Дискретное логарифмирование: пусть $f = \langle a^m \rangle_n$, найти m, зная n, f, a. Для простого n дискретный логарифм называется также ***индексом***.

3) Извлечение корня по модулю: пусть $f = \langle a^m \rangle_n$, найти a, зная n, f, m.

Замечание (об извлечении корня по модулю). Для последней задачи точно известно, что существует эффективный алгоритм, находящий корень по модулю или устанавливающий, что такого корня нет. (См. [1]). Сам эффективный алгоритм не построен. Выяснено, что для его построения достаточно знать разложение n на простые множители. Таким образом, используемых односторонних функций, собственно, две, и современные криптосистемы почти все основаны на проблеме факторизации целых чисел или на проблеме дискретного логарифма в конечной абелевой группе. Задачи эти в настоящее время одинаково трудны.

2. RSA — самая известная криптосистема с открытым ключом

На первой из этих односторонних функций основана предложенная в 1978 г. и выдержавшая испытание временем криптосистема RSA (Rivest, Shamir, Adleman [2]).

Вот её краткое описание.

Представим себе сеть абонентов, где каждые два должны иметь возможность обмениваться секретной информацией.

Каждый из абонентов сети

(1) выбирает 2 различных простых числа p и q;

(2) находит $n = p\,q$ и функцию Эйлера $\varphi(n) = (p-1)(q-1)$;

(3) выбирает целое число e такое, что $e < \varphi(n)$ и $\text{НОД}\big(e, \varphi(n)\big) = 1$;

(4) числа n и e помещает в открытый справочник (это ***открытый ключ***).

Теперь любой абонент сети может послать данному абоненту секретное сообщение с помощью открытого ключа: сообщение представляется в виде целого числа $a < n$ (длинное сообщение разбивают на блоки) и шифруется по формуле $b = f(a) = \langle a^e \rangle_n$. Прочесть его можно, зная целое d такое, что $\langle e\,d \rangle_{\varphi(n)} = 1$ (обратный вычет)[1]. Обозначения e и d являются традиционными и происходят от *encode* и *decode*. При известном d декодирование производится по формуле $a = \langle b^d \rangle_n$.

Обоснование этой формулы:

Покажем, что для любого целого a от 0 до $n-1$ выполнится соотношение

$$\Big\langle \big(\langle a^e \rangle_n\big)^d \Big\rangle_n = a\,.$$

Действительно,

$$\Big\langle \big(\langle a^e \rangle_n\big)^d \Big\rangle_n = \langle a^{ed} \rangle_n\,.$$

Так как e и d взаимно обратны по модулю $\varphi(n)$, то $e\,d = 1 + k\,\varphi(n)$ для некоторого натурального k, поэтому

$$\langle a^{ed} \rangle_n = \langle a^{1+k\,\varphi(n)} \rangle_n = \langle a \rangle_n \cdot \langle a^{\varphi(n)} \rangle_n^k = \langle a \rangle_n = a\,.$$

Использована теорема Эйлера $\langle a^{\varphi(n)} \rangle_n = 1$.

Закодированный с помощью RSA текст защищён от несанкционированного прочтения настолько, насколько затруднено разложение на множители числа n.

В связи с этим развиваются алгоритмы разложения целых чисел на множители. В известном конкурсном списке RSA [3] приводятся большие числа, за факторизацию которых объявлены премии. Выпишем здесь для наглядности два из этих чисел:

- число RSA-640 (640 двоичных, 193 десятичных цифры, уже факторизовано):

  3107418240490043721350750035888567930037346022842727545720161948823206440518081504556346829 6a2328678243791627283803341547107310850191954852900733772482278352574238645401469173660247765234 6609;

- число RSA-704 (704 двоичных, 212 десятичных цифр, не факторизовано, премия \$30000):

  7403756347956171282804679609742957314259318888923128908493623263897276503402826627689199641962511784399589433050212758537011896809828673317327310893090055250511687706329907239638078671008609696253793465056379 6359.

[1] Значение d можно определить, зная p и q: находим $\varphi(n) = (p-1)(q-1)$ и затем вычисляем d по расширенному алгоритму Евклида.

3. Выбор темы — алгоритм квадратичного решета

В то время, когда был изобретён RSA, числа более чем с 80 десятичными знаками не поддавались разложению. Все известные алгоритмы либо работали слишком медленно, либо требовали чисел специального вида. Это делало относительно безопасными даже маленькие, 256-битовые ключи. Первым серьёзным прорывом было квадратичное решето, quadratic sieve (QS). Это относительно простой алгоритм факторизации, предложенный Carl Pomerance в 1981 г., который может разлагать на множители числа до 110 десятичных разрядов или около того и для таких чисел остается лучшим. Для чисел ещё больших применяется метод решета числового поля, general number field sieve (GNFS). Однако метод решета числового поля даже для базового описания требует сложных многосторонних разъяснений и обоснований. В то же время основные идеи обоих методов решета совпадают.

4. Основные идеи

4.1. Метод Ферма: факторизация, использующая разность квадратов

Известный из алгебры метод Ферма состоит в вычислении квадратов по модулю n для целых x, чуть больших $\sqrt{n}$, в надежде встретить полный квадрат y^2. Метод быстро работает, если $n = p\,q$ и числа p, q близки друг другу. (Вот почему в RSA не выбирают близких чисел p, q.)

Суть метода. Пусть надо разложить на множители число n. Если удастся найти два числа x и y такие, что $x^2 - y^2 = n$, то $(x + y)(x - y) = n$.

Числа $(x+y)$ и $(x-y)$ являются множителями n, возможно, тривиальными (когда одно из этих чисел 1, а другое n.)

Эти два числа x и y, дающие $x^2-y^2 = n$, найдутся, если найдётся такое целое x, что x^2-n является квадратом. Тогда $x^2-(x^2-n)$ — разность квадратов, равная n.

Поиск начинают с $x = \lfloor\sqrt{n}\rfloor + 1$, наименьшего возможного числа, при котором разность $x^2 - n$ положительна. Увеличивают x на 1 и вычисляют $x^2 - n$, пока $x^2 - n$ не окажется точным квадратом. Если это произошло, пытаются разложить n как $\left(x - \sqrt{x^2 - n}\right)\left(x + \sqrt{x^2 - n}\right)$. Если это разложение тривиально, продолжают увеличивать x.

В качестве примера продемонстрируем разложение на множители числа $n = 364729$. Найдём $\lfloor\sqrt{n}\rfloor + 1 = 604$. Вычисляем

$$604^2 - 364729 = 87 \quad \text{— не точный квадрат.}$$

Продолжаем:

$$605^2 - 364729 = 1296 = 362 \quad \text{— точный квадрат.}$$

Тогда

$$364729 = 605^2 - 36^2 = (605 + 36)(605 - 36) = 641 \cdot 569\,.$$

Разумеется, для очень больших n нет оснований считать, что искомый полный квадрат найдётся случайно возле $\sqrt{n}$.

Ключевым для дальнейшего продвижения явилось наблюдение, что если мы возьмём несколько значений $x^2 - n$, ни одно из которых само не является квадратом, и перемножим их, то мы можем получить квадрат — скажем, y^2. Пусть X — произведение $X = \prod x_i$ таких x_i, для которых $\prod(x_i^2 - n) = y^2$. Тогда $X^2 - y^2$ кратно n. Следовательно, $(X - y)(X + y)$ — факторизация *некоторого кратного* n; другими словами, по крайней мере одна из скобок делится на какой-то множитель n. Вычисляя НОД каждой скобки и n по алгоритму Евклида, мы можем выявить этот множитель. Он опять может быть тривиальным; в таком случае нам не повезло и надо попытаться снова с другим квадратом.

Уточним понятие «везения». Мы проверяем для найденных целых X и $y = \sqrt{X^2 - n}$ условие нетривиальности

$$1 < \text{НОД}(n, X \pm y) < n\,,$$

которое можно записать также $\langle X \pm y \rangle_n \neq 0$ или $\langle X \rangle_n \neq \langle \pm y \rangle_n$.

При выполнении этого условия получаем, что НОД$(n, X + y)$ — множитель n и НОД$(n, X - y)$ — множитель n.

Это условие выполняется с вероятностью 0.5, если $n = p\,q$, и с ещё большей вероятностью для других n. Это следует из теории квадратичных вычетов.

> Подробнее об этом.
>
> Вычет a (целое число от 0 до $n - 1$, или, что является более предпочтительным для излагаемого алгоритма, от $-n/2$ до $n/2$), называют квадратичным вычетом по модулю n, если a и n взаимно просты и существует такое x, что $\langle x^2 \rangle_n = a$. В противном случае a — квадратичный невычет. Пусть a — квадратичный вычет по модулю $n = p\,q$, p и q — различные простые числа. Известно, что в этом случае
>
> - $\langle a \rangle_p$ — квадратичный вычет по модулю p;
> - $\langle a \rangle_q$ — квадратичный вычет по модулю q;
> - $\langle a \rangle_p$ имеет два корня по модулю p, x и $-x$, в интервале от $-n/2$ до $n/2$ (или x и $p - x$ в интервале от 0 до $n - 1$);

- $\langle a \rangle_q$ имеет два корня по модулю q, y и $-y$ (или y и $p - y$);
- любая комбинация корня по модулю p и корня по модулю q позволяет найти один корень по модулю n.

Возможных комбинаций четыре. Вероятность получить нетривиальные множители в этом случае 0.5.

Осталось выяснить, как из данного списка значений $x^2 - n$ выделить такие $x_i^2 - n$, произведение которых — квадрат.

4.2. Нахождение подмножества из списка целых, произведение которых — квадрат

Пусть дан набор целых чисел и необходимо выявить такое его подмножество, чтобы произведение чисел этого подмножества было точным квадратом, если это возможно.

Например, дан набор $\{10, 24, 35, 52, 54, 78\}$, произведение $24 \times 52 \times 78$ равно $97344 = 312^2$. Грубая попытка перебрать все подмножества слишком трудоёмка. Мы используем другой подход, основанный на разложении чисел на простые множители.

Вначале разложим каждое число набора (считаем, что это легко):

$$
\begin{aligned}
10 &= 2 \times 5\,, \\
24 &= 2^3 \times 3\,, \\
35 &= 5 \times 7\,, \\
52 &= 2^2 \times 13\,, \\
54 &= 2 \times 3^3\,, \\
78 &= 2 \times 3 \times 13\,.
\end{aligned}
$$

Перемножая два каких-то числа, мы просто складываем показатели степеней используемых простых чисел. Произведение будет квадратом, если и только если все степени простых чисел, его составляющие, будут чётными. Запишем каждое выше полученное разложение в виде вектора показателей, где k-я компонента показывает степень k-го простого числа:

$$
\begin{aligned}
&\begin{pmatrix} 1 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}, \\
&\begin{pmatrix} 3 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}, \\
&\begin{pmatrix} 0 & 0 & 1 & 1 & 0 & 0 \end{pmatrix}, \\
&\begin{pmatrix} 2 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \\
&\begin{pmatrix} 1 & 3 & 0 & 0 & 0 & 0 \end{pmatrix}, \\
&\begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 1 \end{pmatrix}.
\end{aligned}
$$

Умножая числа, мы просто складываем векторы. Так как нас интересует только чётность, берём все компоненты по модулю 2:

$$\begin{pmatrix}1 & 0 & 1 & 0 & 0 & 0\end{pmatrix},$$
$$\begin{pmatrix}1 & 1 & 0 & 0 & 0 & 0\end{pmatrix},$$
$$\begin{pmatrix}0 & 0 & 1 & 1 & 0 & 0\end{pmatrix},$$
$$\begin{pmatrix}0 & 0 & 0 & 0 & 0 & 1\end{pmatrix},$$
$$\begin{pmatrix}1 & 1 & 0 & 0 & 0 & 0\end{pmatrix},$$
$$\begin{pmatrix}1 & 1 & 0 & 0 & 0 & 1\end{pmatrix}.$$

Нам нужно получить нетривиальную линейную комбинацию по модулю 2 (с коэффициентами 0 и 1), дающую нулевой вектор $\begin{pmatrix}0 & 0 & 0 & 0 & 0 & 0\end{pmatrix}$.

Это можно перефразировать как матричное уравнение $A\,x = \mathbf{0}$ (умножение и сложение по правилам для вычетов по модулю 2), где неизвестным является вектор коэффициентов линейной комбинации, а матрица A состоит из векторов показателей, записанных по столбцам:

$$\begin{pmatrix}1 & 1 & 0 & 0 & 1 & 1\\ 0 & 1 & 0 & 0 & 1 & 1\\ 1 & 0 & 1 & 0 & 0 & 0\\ 0 & 0 & 1 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 1 & 0 & 1\end{pmatrix}.$$

Вектор $x = \begin{pmatrix}0 & 1 & 0 & 1 & 0 & 1\end{pmatrix}$, как видим, является нетривиальным решением уравнения $A\,x = \mathbf{0}$ и соответствует приведённому выше произведению $24 \times 52 \times 78$.

Обсудим способ получения нетривиальных решений.

То, что матричное уравнение $A\,x = \mathbf{0}$ дано над полем вычетов по модулю 2, упрощает алгоритм гауссова исключения неизвестных. В частности, из элементарных преобразований требуется только прибавление к строке другой строки по модулю 2 (булевская операция $\oplus$).

Опишем простой алгоритм на псевдоязыке. Матрицу A будем приводить к виду, позволяющему сразу выписать общее решение системы. Массивы A[1..m, 1..n] и used[1..m] (последний заведён для пометок строк) отнесём к булевскому типу, но сохраним для их элементов обозначения 1 и 0 вместо **true** и **false**. Также используем для логических операций и отношений знаки $\wedge$, $\neg$, $\neq$, $\oplus$ вместо **and**, **not**, <> и **xor**.

Выполним действия:

```
for r := 1 to m do used[r] := 0;
for k := 1 to n do
for r := 1 to m do
if (A[r,k] ∧ ¬used[r]) then begin
  for i := 1 to m do if (A[i,k] ∧ r ≠ i then
     for j := 1 to n do A[i,j]:= A[i,j] ⊕ A[r,j];
  used[r] := 1;
end;
```

Когда описанный алгоритм отработает, каждая строка матрицы, кроме полностью нулевых, будет начинаться с ***ведущей единицы*** (элемента строки, равного 1 и стоящего в столбце с наименьшим номером), и при этом позиции всех ведущих единиц будут различны.

Например, матрица

$$\begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \end{pmatrix}$$

примет вид

$$\begin{pmatrix} 0 & \boxed{1} & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & \boxed{1} & 0 & 1 & 1 \\ \boxed{1} & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \boxed{1} & 1 & 0 \end{pmatrix}.$$

Ведущие единицы выделены.

Результат будет более наглядным, если отсортировать строки по возрастанию позиций ведущих единиц, а последними поместить нулевые строки, не получившие пометку *used*. Тогда матрица примет ***ступенчатый вид***:

$$\begin{pmatrix} \boxed{1} & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & \boxed{1} & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & \boxed{1} & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & \boxed{1} & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix},$$

то есть такой, что для всех $i > 1$ ведущая единица в i-й строке находится хотя бы на одну позицию правее, чем в $(i-1)$-й.

Число ведущих единиц равно рангу матрицы A. Система уравнений заведомо имеет нетривиальные решения, если этот ранг меньше числа столб-

цов (в частности, если число строк m меньше числа столбцов n)[2].

Пусть нетривиальные решения имеются. Нам достаточно найти одно из них.

В соответствии с наличием или отсутствием ведущих единиц в соответствующих столбцах переменные разделились на ***базисные*** и ***свободные***[5].

Теперь мы берём любой столбец, отвечающий свободной переменной, и эту переменную полагаем равной 1. Остальные свободные переменные можем брать как 0, так и 1. Итоговая матрица в строках с ведущими единицами уже содержит выражения базисных переменных через свободные, так что и их мы найдём.

В последнем примере нарочно взят один нулевой столбец, чтобы показать, что эта экзотическая ситуация ничего не меняет в алгоритме. Напротив, мы можем сразу взять значение $x_3 = 1$, остальные свободные переменные положить равными 0, тогда и все базисные переменные будут равны 0. Решение $\begin{pmatrix}0 & 0 & 1 & 0 & 0 & 0 & 0\end{pmatrix}$ сообщает, что третье число из набора, давшего линейную систему, является полным квадратом.

Есть и ещё решения; например, при значениях базисных переменных $x_3 = 1, \quad x_6 = 1, \quad x_7 = 0$ из первой строки находим $x_1 = x_6 = 1$, из второй $x_2 = x_7 = 0$, из третьей $x_4 = x_6 \oplus x_7 = 1$, из четвёртой $x_5 = x_6 = 1$. Решение $\begin{pmatrix}1 & 0 & 1 & 1 & 1 & 1 & 0\end{pmatrix}$ означает, что произведение 1-го, 3-го, 4-го, 5-го и 6-го чисел набора — полный квадрат.

Применим этот метод к приводившейся выше системе уравнений, построенной для набора $\{10, 24, 35, 52, 54, 78\}$. Метод преобразует матрицу

$$\begin{pmatrix}1 & 1 & 0 & 0 & 1 & 1\\ 0 & 1 & 0 & 0 & 1 & 1\\ 1 & 0 & 1 & 0 & 0 & 0\\ 0 & 0 & 1 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 1 & 0 & 1\end{pmatrix}$$

к виду

$$\begin{pmatrix}1 & 0 & 0 & 0 & 0 & 0\\ 0 & 1 & 0 & 0 & 1 & 1\\ 0 & 0 & 1 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 1 & 0 & 1\end{pmatrix}.$$

[2]В случае, когда ранг матрицы A равен числу столбцов, нетривиальных решений не будет. На практике это означает, что исследуемых целых чисел набрано недостаточно.

Двигаясь по строчкам сверху вниз, выписываем общее решение

$$x_1 = 0\,, \quad x_2 = x_5 \oplus x_6\,, \quad x_3 = 0\,, \quad x_4 = x_6\,.$$

Нетривиальных решений три: при $x_5 = 0$, $x_6 = 1$ уже известное нам $\begin{pmatrix}0 & 1 & 0 & 1 & 0 & 1\end{pmatrix}$; при $x_5 = 1$, $x_6 = 0$ решение $\begin{pmatrix}0 & 1 & 0 & 0 & 1 & 0\end{pmatrix}$, соответствующее произведению $24 \times 56 = 1296 = 36^2$; при $x_5 = 1$, $x_6 = 1$ решение $\begin{pmatrix}0 & 0 & 0 & 1 & 1 & 1\end{pmatrix}$, соответствующее произведению $52 \times 54 \times 78 = 219024 = 468^2$.

Осталась проблема: если среди наших чисел есть число с очень большими множителями, то в матрице будет очень много строк, и наш метод станет неэффективным. Во избежание этого мы требуем, чтобы входные числа были ***B-гладкими***, что означает,что их простые множители не превышают некоторого целого B. Если говорят просто о гладких числах, значит, имеют в виду. что B весьма мало в сравнении с n.

Нам предстоит применять этот метод для чисел вида $x^2 - n$ при $x = \lfloor\sqrt{n}\rfloor + 1, \lfloor\sqrt{n}\rfloor + 2, \ldots$. Что делать, если в разложении этих чисел присутствуют большие простые множители?

Мы просто отбросим числа с большими множителями! Теоретические исследования показывают, что имеется достаточно большое количество значений в последовательности $x^2 - n$, которые являются гладкими.

Например, рассмотрим число $n = 112093$. Здесь $\sqrt{n} \approx 334.8$.

Начнём с $x = \lfloor\sqrt{n}\rfloor + 1 = 335$ и будем увеличивать x до $M = 350$. Зададим предварительно границу гладкости $B = 29$. Выбор чисел M и B определяется следующими соображениями:

- чем меньше значения M и B, тем быстрее работает алгоритм;
- значения M и B должны быть достаточно велики, чтобы описанная выше однородная система линейных уравнений по модулю 2 имела нетривиальное решение.

Таким образом, по ходу решения мы сможем откорректировать выбранные значения как в сторону увеличения, так и в сторону уменьшения.

Мы получили значения:

x	335	336	337	338	339	340	341	342	343	344
$x^2 - n$	132	803	1476	2151	2828	3507	4188	4871	5556	6243

x	345	346	347	348	349	350	...	374	375
$x^2 - n$	6932	7623	8316	9011	9708	10407	...	27783	28352

Ни одно из значений вида $x^2 - n$ не квадрат; однако, если мы разложим каждое значение по степеням простых чисел, мы увидим, что четыре из них не имеют множителей больше 11:

$$\begin{aligned} 132 &= 2^2 \times 3 \times 11\,, \\ 7623 &= 3^2 \times 7 \times 11^2\,, \\ 8316 &= 2^2 \times 3^3 \times 7 \times 11\,, \\ 27783 &= 3^4 \times 7^3. \end{aligned}$$

(Позже покажем, как эти четыре числа найти, не раскладывая весь список на множители.)

Соответствующая система уравнений $A\,x = \mathbf{0}$ имеет матрицу

$$A = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix}.$$

Решением её по модулю 2 будет вектор $\begin{pmatrix} x_3 & x_3 + x_4 & x_3 & x_4 \end{pmatrix}$.

При $x_3 = 1, x_4 = 0$ это решение нетривиально: $\begin{pmatrix} 1 & 1 & 1 & 0 \end{pmatrix}$. Это значит, что точным квадратом будет произведение первых трёх чисел:

$$2^2 \cdot 3 \cdot 11 \cdot 3^2 \cdot 7 \cdot 11^2 \cdot 2^2 \cdot 3^3 \cdot 7 \cdot 11 = 2^4 \cdot 3^6 \cdot 7^2 \cdot 11^4 = (2^2 \cdot 3^3 \cdot 7 \cdot 11^2)^2 = 91476^2\,.$$

В обозначениях предыдущего пункта $X = \prod x_i = 335 \cdot 346 \cdot 347$, $y = 91476$. Здесь $X - y = 40220770 - 91476 = 40129294$. Находим

$$\text{НОД}(X - y, n) = \text{НОД}(40129294, 112093) = 112093\,.$$

К сожалению, найденный делитель числа $n = 112093$ тривиален.

Заметим, что и $\text{НОД}(X + y, n) = \text{НОД}(40312246, 112093) = 1$ — тривиальный делитель.

Возьмем тогда другое решение системы $A\,x = \mathbf{0}$, а именно решение $\begin{pmatrix} 0 & 1 & 0 & 1 \end{pmatrix}$, получающееся при $x_3 = 0, x_4 = 1$. Оно означает, что точным квадратом будет произведение второго и четвёртого чисел:

$$3^2 \cdot 7 \cdot 11^2 \cdot 3^4 \cdot 7^3 = 3^6 \cdot 7^4 \cdot 11^2 = (3^3 \cdot 7^2 \cdot 11)^2 = 14553^2\,.$$

Имеем $X = \prod x_i = 346 \cdot 374 = 129404$, $y = 14553$. Находим

$$\text{НОД}(X - y, n) = \text{НОД}(114851, 112093) = 197\,.$$

Найден нетривиальный делитель 197 числа $n = 112093$.

Теперь число n можно разложить на множители: $112093 = 569 \cdot 197$.

4.3. Факторные базы

Имеется ряд методов, основанных на изложенном выше принципе, в том числе и методы решета. Они пользуются следующей терминологией.

Набор небольших простых чисел, к которому обычно добавляют и -1, образует так называемую ***факторную базу*** $\{-1, p_1, p_2, \ldots, p_h\}$, где p_i — небольшие простые числа, ограниченные некоторым B, $p_i \leqslant B$.

Добавление -1 вызвано тем, что в этих методах экономнее работать не с наименьшими положительными, а с **наименьшими по абсолютной величине** вычетами по модулю n, но в нашем упрощённом изложении мы этого не требуем. Мы работали в нашем примере с факторной базой $\{2, 3, 7, 11\}$, хотя первоначально планировалась граница $B = 29$. (Оказалось, что и базы $\{2, 3, 7, 11\}$ достаточно для решения системы).

Обращает на себя внимание отсутствие в факторной базе малого простого числа 5. Действительно, в факторную базу включают только такие простые p_i, для которых $(\frac{n}{p_i}) = 1$ (символ Лежандра; его равенство 1 означает, что n — квадратичный вычет по модулю p_i). Если это не так, то среди чисел вида $x^2 - n$ не найдётся ни одного, делящегося на p_i, и это число не имеет смысла включать в базу.

Символ Лежандра для простого нечётного p вычисляется по формуле $(\frac{n}{p}) = \langle n^{\frac{p-1}{2}} \rangle_p$, вычисление несложно. Число n оказалось квадратичным невычетом по модулю 5, а также по модулям $13, 17, 19, 23, 29, 37, \ldots$, так что если бы границы $B = 11$ оказалось недостаточно, пришлось бы расширять базу, включая в неё простые числа $31, 41, 43$ и так далее.

4.4. Просеивание

Пора вернуться к узкому месту предыдущих выкладок: как из набора чисел вида $x^2 - n$ выделить гладкие числа, не раскладывая каждое на множители.

Основная идея: вместо того, чтобы перебирать эти числа и проверять, делятся ли их квадраты по модулю n на простые числа из базы, перебирают по очереди простые числа из базы и сразу для всех чисел вида $x^2 - n$ проверяют, делятся ли они на очередное p и его степени.

Метод основан на решете Эратосфена, откуда и получил своё название. Решетом служат простые числа факторной базы и дополнительно их небольшие степени, и при просеивании мы числа не вычёркиваем, а **делим на эти простые**. Те числа, что перейдут в 1, будут B-гладкими.

Продемонстрируем этот метод на нашем наборе чисел:

132	803	1476	2151	2828	3507	4188	4871	5556
6243	6932	7623	8316	9011	9708	10407	11108	11811
12516	13223	13932	14643	15356	16071	16788	17507	18228
18951	19676	20403	21132	21863	22596	23331	24068	24807
25548	26291	27036	27783	28532				

Начинаем с 2. Первое просеивание:

66	803	738	2151	1414	3507	2094	4871	2778
6243	3466	7623	4158	9011	4854	10407	5554	11811
6258	13223	6966	14643	7678	16071	8394	17507	9114
18951	9838	20403	10566	21863	11298	23331	12034	24807
12774	26291	13518	27783	14266				

Заметим, что в исходной таблице чётные числа шли через одно. Это, конечно, не случайность. Школьная алгебра показывает, что для любого целого d выполняется равенство

$$(x + k\,d)^2 - n = (x^2 - n) + d\,(2\,k\,x + d\,k^2)\,,$$

так что если число из нашего набора делится на d, то и все числа, отстоящие от него на расстояние, кратное d, делятся на d.

Есть изменённые числа (в таблице они выделены). Снова прогоняем через 2:

33	803	369	2151	707	3507	1047	4871	1389
6243	1733	7623	2079	9011	2427	10407	2777	11811
3129	13233	3483	14643	3839	16071	4197	17507	4557
18951	4919	20403	5283	21863	5649	23331	6017	24807
6387	26291	6759	27783	7133				

При следующем прогоне через делитель 2 изменений уже не будет (в данном маленьком примере это видно до прогона), так что дальше делим уже на 3 — следующее простое число факторной базы:

11	803	123	717	707	1169	349	4871	463
2081	1733	2541	693	9011	809	3469	2777	3937
1043	13223	1161	4881	3839	5357	1399	17507	1519
6317	4919	6801	1761	21863	1883	7777	6017	8269
2129	26291	2253	9261	7133				

Нас уже не удивляет, что если число делится на 3, то и каждое третье после него делится на 3. Не должно удивлять нас и то, что таких цепочек две, одна начинается с 1-й клетки, другая с 3-й. Наличие двух цепочек вытекает из того, что квадратичное сравнение по модулю p имеет ровно два решения (если вообще имеет решение). Разумеется, это наблюдение, сильно сокращающее расчёты, используется при просеивании: находят первые два числа, делящиеся на p, и идут от них «с периодом p». Для малых простых можно обойтись без специального алгоритма, просто тестировать первые p чисел, чтобы определить, какие из них делятся на p, а потом идти по цепочке. Для больших простых такой трудоёмкой процедуры следует избегать. Существует эффективный алгоритм для нахождения этих двух решений (Shanks-Tonelli algorithm [5]).

Повторяем деление на 3 ещё 3 раза и получаем:

11	803	41	239	707	1169	349	4871	463
2081	1733	847	77	9011	809	3469	2777	3937
1043	13223	43	1627	3839	5357	1399	17507	1519
6317	4919	2267	587	21863	1883	7777	6017	8269
2129	26291	751	343	7133				

При следующем прогоне через делитель 3 изменений не будет. Дальше делим на 7 — следующее число факторной базы:

11	803	41	239	101	167	349	4871	463
2081	1733	121	11	9011	809	3469	2777	3937
149	1889	43	1627	3839	5357	1399	2501	217
6317	4919	2267	587	21863	269	1111	6017	8269
2129	26291	751	49	1019				

Попутно приятно обратить внимание на две цепочки делителя 7, с 4-й и с 5-й клетки.

Ещё два прогона через 7 приводят к первой единице:

11	803	41	239	101	167	349	4871	463
2081	1733	121	11	9011	809	3469	2777	3937
149	1889	43	1627	3839	5357	1399	2501	31
6317	4919	2267	587	21863	269	1111	6017	8269
2129	26291	751	1	1019				

При следующем прогоне через 7 изменений не будет. Делим на 11. Это понадобится проделать дважды, после чего получим числа:

1	73	41	239	101	167	349	4871	463
2081	1733	1	1	9011	809	3469	2777	3937
149	1889	43	1627	349	487	1399	2501	31
6317	4919	2267	587	21863	269	101	547	8269
2129	26291	751	1	1019				

Появились новые единицы, и эта таблица для нас последняя.

Там, где теперь стоят 1, в исходной таблице находились 11-гладкие числа.

Их список мы уже приводили: $132, 7623, 8316, 27783$. Разложение их на множители получено параллельно с просеиванием.

5. Заключительные замечания

Мы изложили базовую реализацию метода квадратичного решета, она имеет много версий. Рекорд метода — разложение RSA-129.

Метод имеет субэкспоненциальную эмпирическую сложность, то есть наихудшее время его работы представимо в форме $e^{o(N)}$, где $N = log\, n$ (число двоичных разрядов в записи n), оценка времени выполнения $O(e^{(C+o(1))\sqrt{NlogN)}})$ с константой $C = 9/8$.

Метод решета в числовом поле имеет лучшую оценку $O(e^{(C+o(1))\sqrt[3]{n}\cdot\sqrt[3]{log^2 N}})$.

Для более детального ознакомления с методами факторизации и их местом в современной криптологии можно рекомендовать книги [1, 6–9].

ЛИТЕРАТУРА

1. Брассар Ж. *Современная криптология.* М.: ПОЛИМЕД, 1999.

2. Rivest R. L., Shamir A., Adleman L. M. *A method for obtaining digital signatures and public-key cryptosystems* // Communications of the ACM. 1978. V. 21, P. 120–126.

3. *The RSA Challenge Numbers* // RSA Laboratories. http://www.rsasecurity.com/rsalabs/node.asp?id=2093

4. Стренг Г. *Линейная алгебра и её применения.* М.: Мир, 1980.

5. Shanks D. *Five number-theoretic algorithms* // Proceedings of the Second Manitoba Conference on Numerical Mathematics (Winnipeg), Utilitas Math. 1973. P. 51–70. Congressus Numerantium, No. VII.

6. Смарт Н. *Криптография.* М.: Техносфера, 2005.

7. Тилборг ван Х. К. А. *Основы криптологии: профессиональное руководство и интерактивный учебник*. М.: Мир, 2006.

8. Земор Ж. *Курс криптографии*. М.-Ижевск: НИЦ «Регулярная и космическая динамика»; Институт космических исследований, 2006.

9. Василенко О. Н. *Теоретико-числовые алгоритмы в криптографии*. М.: МЦНМО, 2003.