Лабораторная работа № 1

Атака на RSA с малым открытым ключом

Сдача: 12.02.24

1 Алгоритм Копперсмита нахождения малых корней многочлена

В основе атаки на одностороннюю функцию RSA лежит следующая теорема, доказанная Доном Копперсмитом в [1].

Теорерма 1. *Положим, N – целое, $f \in \mathbb{Z}[x]$ – унитарный многочлен степени n. Положим далее, $X = N^{\frac{1}{n}-\varepsilon}$ для $\varepsilon > 0$. Тогда существует алгоритм, который вернет все $|x_0| < X$, удовлетворяющие $f(x_0) = 0 \bmod N$, за время, равному времени работы алгоритма LLL на решетки размерности $\mathcal{O}(\min\{\frac{1}{\varepsilon}, \log_2 N\})$.*

Прелесть это теоремы состоит в том, что модуль N может быть составным числом (для простых модулей необходимости в использовании теоремы Копперсмита нет, так как существуют более быстрые алгоритмы нахождения корней).

Далее мы докажем эту Теорему 1. Начнем с результата, полученным Хогрейв-Хрэхэмом [2]. Многочлену $h(x) = \sum_{i=0}^{n} a_i x^i \in \mathbb{Z}[x]$ будем сопоставлять вектор-коэффициентов $(a_i)_i \in \mathbb{Z}^{n_1}$ и определять квадрат нормы $\|h\|^2 = \sum_i |a_i|^2$.

Лемма 2. *Пусть $h(x) \in \mathbb{Z}[x]$ – многочлен степени n и $X > 0$ – целое. Положим, $\|h(xX)\| < N/\sqrt{n}$. Если $|x_0| < X$ удовлетворяет $h(x_0) = 0 \bmod N$, то уравнение $h(x_0) = 0$ выполняется над $\mathbb{Z}$.*

Доказательство.

$$\begin{aligned}|h(x_0)| = \left|\sum_i a_i x_0^i\right| = \left|\sum_i a_i X^i \left(\frac{x_0}{X}\right)^i\right| \le \sum_i \left|a_i X^i \left(\frac{x_0}{X}\right)^i\right| \\ < \sum_i \left|a_i X^i\right| \le \sqrt{n}\,\|h(xX)\| < N.\end{aligned}$$

Из этого неравенства и условия $h(x_0) = 0 \bmod N$, следует $h(x_0) \equiv 0$. □

Лемма 2 утверждает, что если h – многочлен малой нормы, то всего его корни $\bmod N$, также малые по абсолютному значению, являются его корнями над целыми числами. Следовательно, мы будем искать для многочлена $f(x)$ (необязательно малой нормы), многочлен $h(x)$ малой нормы, имеющий такие же корни, как $f(x)$. Очевидно, мы могли бы искать линейные комбинации многочленов вида $f, xf, x^2 f, \ldots$, дающие малую норму. Однако, часто такие многочлены не дают желаемую нетривиальную линейную комбинацию. Поэтому Копперсмит предлагает добавлять в список многочленов степени $f(x)$, заметив, что если $f(x) = 0 \bmod N$, то $f(x)^i = 0 \bmod N^i$ для любого $i > 1$. В общем случае зададим для некоторого целого m[1] многочлены

$$g_{i,j}(x) = N^{m-i} x^j f(x)^i, \quad \text{для } i = 0, \ldots, m-1, j = 0, \ldots n-1.$$

[1] Более точный анализ показывает, что $m = \lceil \frac{1}{n\varepsilon} \rceil$, на практике m выбирают небольшой константой.

Тогда x_0 – корень многочлена $g_{i,j}(x)$ по модулю N^m для всех $i \geq 0$. Теперь мы будем искать многочлен $h(x)$ – линейную комбинацию многочленов $g_{i,j}(x)$, такую, что норма $h(xX)$ не превосходит N^m (выбор многочленов $g_{i,j}(xX)$ позволяет увеличить границу с N до N^m).

Решим задачу поиска линейной комбинации с малой нормой. Сопоставляя многочленам $g_{i,j}(xX)$ вектора, составленные из их коэффициентов, задача поиска $h(x)$ сводится к поиску короткого вектора в решетке, образованной матрицей-коэффициентов, где в i-м столбце записан коэффициенты многочленов при i-ой степени x. Получим решетку размерности $w = nm$, базисом которой будет нижне-треугольная матрица (упорядочивая сначала по i, потом по j). Например, для $n = 2, m = 3$ матрица будет иметь вид

$$\begin{array}{c c}
 & \begin{array}{cccccc} x^0 & x^1 & x^2 & x^3 & x^4 & x^5 \end{array} \\
\begin{array}{c} g_{0,0}(xX) \\ g_{0,1}(xX) \\ g_{1,0}(xX) \\ g_{1,1}(xX) \\ g_{2,0}(xX) \\ g_{2,1}(xX) \end{array} &
\begin{pmatrix}
N^3 & & & & & \\
\star & N^3X & & & & \\
\star & \star & N^2X^2 & & & \\
\star & \star & \star & N^2X^3 & & \\
\star & \star & \star & \star & NX^4 & \\
\star & \star & \star & \star & \star & NX^5
\end{pmatrix}
\end{array}$$

Позиции $\star$ соответствуют коэффициентам многочленов $g_{i,j}(xX)$, пустые позиции соответствуют нулям. Алгоритм LLL, запущенный для этого базиса (здесь базис задан векторами-строками, как в FPyLLL/Sage!), вернет вектор v решётки, чья норма будет удовлетворять $\|v\| \leq 2^w \det(L)^{1/w}$. Определитель решетки можно оценить как[2]

$$\det(L) = \prod_{i=0}^{m-1} N^{(m-i)n} \prod_{j=0}^{n-1} \prod_{i=0}^{m-1} X^j X^{ni} = \prod_{i=1}^{m} N^{in} \prod_{i=0}^{nm-1} X^i =$$
$$= N^{\frac{m(m+1)n}{2}} X^{\frac{mn(mn-1)}{2}} \approx N^{\frac{m^2 n}{2}} X^{\frac{m^2 n^2}{2}}.$$

Для того, чтобы вектор v (соответствующий многочлену $h(xX)$), полученный из алгоритма LLL удовлетворял условию Леммы 2, необходимо выполнение неравенства

$$2^w \det(L)^{1/w} < \frac{N^m}{\sqrt{w}}.$$

Подставляя полученную аппроксимацию для $\det(L)$ и пренебрегая малыми множителями, условие выше дает

$$\det(L) \leq N^{mw} \iff X \leq N^{1/n},$$

что соответствует границе в Теореме 1 в точности до ε, возникающим вследствие аппроксимаций.

2 При чём тут RSA?

2.1 Стереотипные сообщения

Схема шифрования и алгоритм подписи RSA основаны на односторонней функции вида $x \mapsto x^e \bmod N$, для некой $e \in \mathbb{Z}_N^\star$ (такое отображение называется “односторонней функцией с потайным входом”[3], так как зная $d = e^{-1} \bmod \phi(N)$ эту функцию можно эффективно обратить. Так *небезопасная* версия шифрования сообщения m, вычисляет шифр-текст $c = m^e \bmod N$.

Для того, чтобы сделать возведение в степень e эффективным, некоторые реализации RSA выбирали $e = 3$.[4] В этой лабораторной вы убедитесь в том, что это плохая идея. Например, если

[2] Множители, ушедшие из-за аппроксимации в формуле ниже, учитываются в ε.

[3] `https://en.wikipedia.org/wiki/Trapdoor_function`

[4] Сегодня все реализации RSA отказались от такой шифрующей экспоненты.

мы шифруем стереотипные сообщения, такие как “ваш пароль на сегодня: XXXXX”, то шифр-текст такого сообщения есть $(S+x)^e \bmod N$, где S – известная часть сообщения “ваш пароль на сегодня: ’, а пароль x – неизвестная. Тогда шифр-текст соответствует многочлену $f(x) = (S+x)^e - c \bmod N$, где неизвестная часть открытого текста x – его корень. Если шифрующая экспонента e мала, алгоритм Копперсмита позволит эффективно найти x, так как размерность решетки будет небольшой.

2.2 Случайная набивка (padding)

Эта атака на RSA была предложена Фрэнклином-Райтером в 1996 году. Положим, открытые сообщения m, m' связны соотношением $m = m' + r$, где r– малое значение (например, если для шифрования i-го сообщения используется так называемая набивка $R_i = i < 2^k$ для “рандомизации” открытого текста, то $c_i = (m \cdot 2^k + i \bmod N)^5$). Тогда для $e = 3$,

$$c = m^3 \bmod N$$
$$c' = (m+r)^3 \bmod N.$$

Зная c, c' и r, можно легко вычислить m (подумайте, как именно, вам это пригодится далее).

Что если мы не знаем r, но знаем, что оно мало? Тогда два шифр-текста c, c' дадут два уравнения

$$m^3 - c = 0 \bmod N$$
$$(m+r)^3 - c' = 0 \bmod N,$$

в которых неизвестными являются m, r. Используя метод результант (классический метод исключения неизвестного из системы, нам в лабораторной понадобится только значение результанты), по переменной получим многочлен от одной переменной r.

$$\mathrm{res}_m(m^3 - c, (m+r)^3 - c') = r^9 + (3c - 3c')r^6 + 3r^3(c^2 + 7cc' + c'^2) + (c - c')^3 \bmod N.$$

Полученный многочлен $f(r) = r^9 + (3c - 3c')r^6 + 3r^3(c^2 + 7cc' + c'^2) + (c - c')^3$ степени 9 имеет своим корнем искомое значение r. Если $r < N^{1/9}$, Теорема 1 вычислит этот корень.

3 Задание к лабораторной

В этой лабораторной вам даны открытый ключ RSA $(N, e = 3)$ и два шифр-текста (c, c') для сообщений (m, m'), связанных неким малым r. Ваша задача – найти r и пару сообщений.

Параметры заданы в файле `lab1_input.txt` по ссылке `https://crypto-kantiana.com/elena.kirshanova/teaching/lattices_2023/lab1_input.txt`.

Для успешной атаки можете использовать $X = \lfloor 0.5 \cdot N^{1/9} \rfloor$, в определении $g_{i,j}$ можете взять $m = 5$.

Список литературы

[1] Don Coppersmith. *Small solutions to polynomial equations, and low exponent RSA vulnerabilities.* Journal of Cryptology, 10:233–260, 1997

[2] Nick Howgrave-Graham *Finding small roots of univariate modular equations revisited.*. Cryptography and Coding, volume 1355 of Lecture Notes in Computer Science, 131–142. Springer-Verlag, 1997.

[5] Очевидно, такой метод рандомизации не является безопасным