Министерство науки и высшего образования Российской Федерации ФГАОУ ВО «Балтийский федеральный университет имени Иммануила Канта»

УТВЕРЖДАЮ
ректор БФУ им. И. Канта
_____ А.А. Фёдоров
«__»_____ 2022 года

Отчет
о научно-исследовательской работе _____ этап
“Алгоритмы нахождения короткого вектора в алгебраических решетках”
(договор на выполнение НИР № 2165 от 8 октября 2021 года)

Исполнитель _____

Согласовано:
Проректор по научной работе
_____ М.В. Дёмин

Руководитель проекта
_____ Е.А. Киршанова

Калининград
2022

Изучение понятия конфигурации в алгоритмах просеивания

А.С. Каренин

25 марта 2022 г.

1 Введение

Современные методы криптоанализа на решётках используют алгоритмы редукции решёток, сложность которых даёт оценку защищённости существующих решётчатых криптосистем. BKZ алгоритм и производные от него Dual BKZ и Slide reduction в процессе работы используют ортогональный базис векторного пространства, в котором находится решётка, который получается при помощи алгоритма ортогонализации Грама-Шмидта. В [DP16] рассмотрен подход к ускорению ортогонализации, работающий за $O(n^2 \log n)$ и подающий на выход так называемое LD-дерево - компактную репрезентацию LDL* разложения матрицы $G = B \cdot B^* = LDL^*$. В дальнейшем эта репрезентация может быть использована для вызова алгоритма нахождения ближайшей плоскости Бабая (Babai nearest plane algorithm).

В результате проведения работ по данному этапу научного исследования был на практике реализован алгоритм, предложенный в статье [DP16]. Первоначальная реализация, приложенная к статье, имела имплементацию только для решёток ранга 1, тогда как данная реализация имеет возможность обрабатывать решётки произвольного ранга и работать с первоначальном входом в виде элементов циклотомического поля при помощи DP вложения, определение которому будет дано в разделе 2.1 данного отчёта.

2 Быстрая ортогонализация при помощи преобразования Фурье

Будем считать известными определённые в отчёте за февраль следующие понятия:

- Решётка $\mathcal{L}$, её базис и формулировка задачи SVP;
- Числовое поле K, эрмитово сопряжение f^* элемента f числового поля, матрица Грама;
- Операторы вектрификации $V_{d/d'}$ и матрификации $M_{d/d'}$.

2.1 DP вложение

Пусть $K = \frac{\mathbb{Q}[x]}{(\Phi_n(x))}$ - циклотомическое поле ($\Phi_n(x)$ - n-й циклотомический многочлен). Пусть также $R = \frac{\mathbb{Q}[x]}{(x^n-1)}$ - кольцо конволюции. Так как $\Phi_n(x)$ делит $x^n - 1$, мы можем ввести DP вложение элементов из поля K в кольцо R. Пусть $e \equiv 1 \pmod{\Phi_n(x)}$ а также $e \equiv 0 \pmod{\Psi(x)}$, где $\Psi(x) = (x^n-1)/\Phi_n(x)$, - элемент кольца R. Поднимемся канонической инъекцией в поле $\mathbb{Q}[x]$ и умножим исходный элемент f на e, после чего

возьмём результат по модулю $\Psi(x)$. Описанное отображение инъективно, а, следовательно, является биекцией поля K на образ этого вложения. Назовём данное вложение DP вложением в честь Лео Дюки и Томаса Преста.

Таким образом, вкладывая элементы из K в R, мы можем применять методы ортогонализации, опирающиеся на аппарат операторов векторизации и матрификации.

2.2 Сравнение работы реализации алгоритма быстрой ортогонализации при помощи быстрого преобразования Фурье с реализацией, предложенной в статье

Важно отметить, что авторами кода, приложенного к вышеупомянутой статье, был реализован случай решёток ранга 1, тогда как в реализации, приложенной к данному отчёту на вход, может подаваться решётка произвольного ранга. Поэтому практическое сравнение времени работы алгоритма имеет смысл только для параметров, которые в состоянии принять алгоритм Дюки и Преста.

Вызовы Babai nearest plane (BNP) алгоритма производились на алгебраических решётках ранга 1, базис которых представляет из себя циркулянтную матрицу. Методология эксперимента заключается в серии из 100 вызовов BNP на случайно выбранной решётке соответствующей размерности и нахождении среднего арифметического времени выполнения вызова. Внизу представлена таблица, показывающая зависимость времени вызова BNP при помощи кода, предложенного в [DP16], имплементации в рамках данного проекта, а также классическая реализация при помощи fpylll.

Размерность решётки	Время t_0 работы оригинального алгоритма, с.	Время t_1 работы данной реализации, с.	Время t_2 работы fpylll, с.	t_1/t_0	t_2/t_1
128	0,002327	0,009220	0,0185	3,9622	2,007
256	0,004352	0,018909	0,0715	4,3450	3,7813
512	0,008704	0,037317	0,1840	4,2875	4,9307
1024	0,018022	0,065554	0,6312	3,6375	9,6287
2048	0.037442	0.150941	2,2129	4,0313	14,6607

Таблица 1: Сравнение времени работы алгоритмов BNP.

Данные из этой таблицы подтверждают предсказанное в вышеупомянутой статье ассимтотическое ускорение BNP алгоритма относительно fpylll реализации в случае подхода с быстрой ортогонализацией с использованием преобразования Фурье. Первым недостатком реализации, предложенной в рамках проведённых работ, является недостаточная оптимизованность программы. Использование того факта, что матрица L в LDL^* декомпозиции нижнетреугольная, а также оптимизация доступа к эмементам матриц и массивов непосредственно в коде являются тем самым замедляющим программу фактором и могут быть достаточно легко использованы для оптимизации предложенного кода. Реализация алгоритма на языке Python доступна по запросу на почту ASKarenin@stud.kantiana.ru.

Пусть мы вызываем BNP при помощи быстрого преобразования Фурье с целевым вектором $v + c_e$, где $v \in \mathbb{Z}^n, c_e \in \mathbb{R}^n$, на решётке $\mathcal{L}(B)$ размерности n. Тогда на выходе этот алгоритм вернёт вектор v тогда и только тогда, когда $V(c_e \cdot B) \in \mathcal{P}(\widetilde{B_0})$, где $\mathcal{P}(\widetilde{B_0})$ - фундаментальный параллелепипед решётки, порожденной базисом $\widetilde{B_0}$, где $B = L \cdot \widetilde{B_0}$ - GSO разложение матрицы B, а V - оператор векторизации $V_{d/1}$. Этот факт даёт возможность протестировать корректность работы предложенного BNP алгоритма - при добавлении умеренного (удовлетворяющего вышеуказанному условию) шума e,

вызов BNP алгоритма должен вернуть в точности вектор v.

В таблице 2 представлены результаты экспериментов, проведённых на базисах размерности d решёток, полученных из антициркулянтных базисов циклотомических решёток размерности $d/2$. Иными словами, вектор размерности $d/2$, ротации которого задают базис циклотомической решётки, отображался при помощи DP вложения в кольцо R, элементами которого являются многочлены (представленные в виде вектора длины d). Эксперимент был проведён на ноутбуке с процессором i7-1165G7, работающим на частоте 2.8 ГГц. Размер системной памяти равен 16 ГиБ.

Размерность $d/2$ NTRU решётки	Время t_0 работы оригинального алгоритма, с.	Время t_1 работы данной реализации, с.	Время t_2 работы fpylll, с.	t_0/t_1	t_2/t_1
128	0,432009	0,0156	0,461228	27,69	39,24
256	0,881033	0,0332	1,809470	26,54	54,50
512	1,744091	0,0755	7,268182	23,10	96,26

Таблица 2: Сравнение времени работы алгоритмов BNP для решёток - вложений из циклотомических полей.

Как видно из таблицы, представленной в рамках данного отчёта, реализация алгоритма BNP для циклотомического случая даёт многократное преимущество по времени выполнения перед обоими конкурентами. Техническим трюком для такого ускорения является тот факт, что при LD вложении половина векторов ожидается быть линейно зависимой с первой половиной. В коде, приложенном к статье [DP16], эта ситуация обрабатывается вычёркиванием линейно-зависимых векторов при помощи вызова довольно дорогой в плане вычислительных ресурсов функции. В основе кода, разработанного для данного отчёта, лежит наблюдение, заключающееся в том, что при проведении LDL* декомпозиции в случае возникновения линейно-зависимого вектора, в матрице D (а, позже, соответственно и в матрице L) на соответствующем месте появляется бесконечность в результате деления на нуль. В таком случае в соответствующую строку матриц записываются нули, как бы исключая такие векторы из рассмотрения.

Корректность выхода алгоритмов была проверена следующим образом: генерировалась решётка $\mathcal{L}$ и вектор $v \in \mathcal{L}$, после чего к нему добавлялся шум e такой, что $||e_i|| < ||b_i^*||/2 \quad \forall i \in [0, d)$, где b_i^* - длины векторов ортогонального базиса Грама-Шмидта. После этого на решётке $\mathcal{L}$ вызывался алгоритм BNP с целевым вектором $v + e$. Если алгоритм работает корректно, то он должен вернуть v, так как вектор $v + e$ лежит в фундаментальном параллелепипеде, сдвинутом на вектор v, что было и критерием проверки корректности его вывода.

3 Заключение

В рамках данного этапа работ был запрограммирован алгоритм, предложенный в статье [DP16]. Его реализация для колец конволюции уступает оригинальному коду в быстродействии, однако данная имплементация неидеальна и в ней есть ещё возможность оптимизации. В дополнение к этому, данная реализация работает с решётками произвольного ранга, что накладывает некоторые ограничения на типы данных, использующиеся для имплементации.

С другой стороны, благодаря техническому трюку, описанному в конце прошлого пункта, удалось добиться весомого ускорения исполнения алгоритма для случая, ко-

гда решётка является вложением решетки меньшей размерности из циклотомического поля K. Однако требуется строгое доказательство корректности применения такого метода.

Полученные результаты можно использовать для построения LDL декомпозиции в компактной форме и ускорения BNP алгоритма для алгебраических решёток. Однако LLL и BKZ алгоритмы разрушают структуру алгебраической решётки, что делает применение данного подхода “в лоб” невозможным. Однако, в случае NTRU решёток, ротации двух линейно-независимых векторов с большой вероятностью сами обладают свойством линейной независимости, что позволяет применять рассматриваемый в данном отчёте алгоритм к двум кратчайшим векторам из BKZ редуцированного базиса.

Список литературы

[DP16] Léo Ducas and Thomas Prest. Fast fourier orthogonalization. In *Proceedings of the ACM on International Symposium on Symbolic and Algebraic Computation*, ISSAC'16, page 191–198, 2016.