Секция 3

МАТЕМАТИЧЕСКИЕ МЕТОДЫ КРИПТОГРАФИИ

УДК 519.7 DOI 10.17223/2226308X/15/11

РАЗРАБОТКА И СРАВНЕНИЕ МОДЕЛЕЙ КВАНТОВОГО ОРАКУЛА ДЛЯ ГИБРИДНОЙ АТАКИ НА ПОСТКВАНТОВЫЕ КРИПТОСИСТЕМЫ, ОСНОВАННЫЕ НА РЕШЁТКАХ[1]

А. О. Бахарев

Для предложенной ранее модели квантового оракула, используемого в гибридном квантово-классическом алгоритме решения задачи нахождения кратчайшего вектора в решётке, получены новые уточнённые оценки числа кубит и глубины схемы. Разработана и проанализирована новая модель квантового оракула, использующая классическую память для хранения списка векторов. Получены верхние оценки сложности реализации атаки на постквантовые криптосистемы, являющиеся финалистами конкурса NIST.

Ключевые слова: *квантовый поиск, криптография с открытым ключом, постквантовая криптография.*

1. Основные понятия и определения

С каждым годом квантовые вычисления развиваются с большей силой. Поэтому возникает необходимость в разработке и анализе криптосистем, которые будут устойчивы к атакам с использованием квантового компьютера. Стойкость многих известных постквантовых криптосистем, основанных на решётках, зависит от сложности решения задачи нахождения кратчайшего вектора в решётке.

В 2016 г. Национальный институт стандартов и технологий США (NIST) объявил конкурс «Post-Quantum Cryptography Competition», по завершении которого будет принят новый — квантово-устойчивый — стандарт асимметричного шифрования. И 22 июля 2020 г. начался третий этап конкурса, финалистами которого являются криптосистемы, основанные на теории решёток и кодах, исправляющих ошибки.

В данной работе рассмотрен подход на основе решёток.

Определение 1. Пусть $u_1, \ldots, u_d \in \mathbb{R}^n$ — линейно независимые векторы и $d \leqslant n$. *Решёткой размерности* d называется множество

$$\Lambda = \Big\{ \sum_{i=1}^{d} b_i u_i : b_i \in \mathbb{Z} \Big\}.$$

Линейно независимая система векторов, порождающая решётку, называется *базисом решётки*.

Определение 2. *Задача нахождения кратчайшего вектора* (SVP) — найти в заданной своим базисом решётке ненулевой вектор, имеющий наименьшую длину.

[1]Работа выполнена при поддержке Математического центра в Академгородке, соглашение с Министерством науки и высшего образования РФ № 075-15-2022-281.

В общем случае SVP является NP-трудной задачей. Стойкость систем, основанных на решётках, зависит от эффективности решения SVP, так как большинство известных атак сводятся к решению этой проблемы.

В 2015 г. опубликована работа [1], в которой представлен гибридный подход к поиску кратчайшего вектора решётки на основе алгоритма GaussSieve [2] (алгоритм 1) — одного из самых эффективных классических алгоритмов.

Алгоритм 1. Алгоритм GaussSieve (D. Micciancio and P. Voulgaris, 2010)

Вход: B — базис решётки.
Выход: v — кратчайший вектор решётки.

Инициализировать пустой неупорядоченный список L и пустой стек S.
Повторять
  получить вектор v из стека (или сгенерировать новый).
  Пока $w \leftarrow$ ПОИСК$\{w \in L : \|v \pm w\| \leqslant \|v\|\}$
    уменьшить v с помощью w $(v \leftarrow v \pm w)$.
  Пока $w \leftarrow$ ПОИСК$\{w \in L : \|w \pm v\| \leqslant \|w\|\}$
    удалить w из листа L;
    уменьшить w с помощью v $(w \leftarrow w \pm v)$;
    добавить w в стек S.
  Если v изменился, **то**
    добавить v в стек S,
  иначе
    добавить v в лист L.
Пока v не станет кратчайшим вектором.
Вернуть вектор v.

Так как длина неупорядоченного списка L целочисленных векторов фиксированной размерности из алгоритма GaussSieve увеличивается экспоненциально с ростом размерности решётки, самой трудозатратной операцией данного алгоритма является функция «ПОИСК», которая осуществляет перебор неупорядоченного списка L для нахождения элемента w, удовлетворяющего условию поиска: $\|v \pm w\| \leqslant \|v\|$ или $\|w \pm v\| \leqslant \|w\|$. В рамках предложенного в [1] подхода ускорение достигается за счёт использования в функции «ПОИСК» квантового алгоритма поиска. Задача, решаемая этим алгоритмом, называется *задачей поиска*. Предполагается, что есть неупорядоченный список из K элементов, и требуется найти один элемент, удовлетворяющий некоторому условию. Другими словами, определена булева функция f, которая по номеру элемента (его двоичному представлению x) определяет, является ли элемент подходящим. Если элемент подходящий, то $f(x) = 1$, иначе $f(x) = 0$. В такой постановке задача поиска сводится к нахождению решения уравнения $f(x) = 1$.

В классическом варианте при условии, что решение одно, требуется $\sim K/2$ обращений к функции f для нахождения решения. Квантовый алгоритм поиска элемента в неупорядоченном списке (алгоритм Гровера [3]) решает данную задачу за $\sim \frac{\pi}{4}\sqrt{K}$ обращений к *оракулу* — квантовому аналогу функции f. Известно, что любая булева функция может быть реализована на квантовом компьютере.

2. Квантовые вычисления

Квантовый компьютер, в отличие от обычного, оперирует *квантовыми битами* (*кубитами*) [4]. Подобно классическому биту, который может находиться в состоянии 0 или 1, кубит имеет возможные состояния $|0\rangle$ и $|1\rangle$. Символ «$|\rangle$» называется *дираковским обозначением*, он является стандартным обозначением состояния в квантовой механике. Различие между битами и кубитами в том, что кубит может находиться в состоянии, отличном от $|0\rangle$ или $|1\rangle$. Можно составить линейную комбинацию состояний (суперпозицию):

$$|\varphi\rangle = \alpha|0\rangle + \beta|1\rangle.$$

Числа α и β являются комплексными и $|\alpha|^2 + |\beta|^2 = 1$. Иначе говоря, состояние одного кубита можно представить как единичный вектор из $\mathbb{C}^2$. Однако мы не можем измерить кубит, чтобы определить его квантовое состояние, т. е. значения α и β. Из квантовой механики следует, что при измерении кубита мы получаем либо результат 0 с вероятностью $|\alpha|^2$, либо результат 1 с вероятностью $|\beta|^2$. Систему с произвольным количеством кубит описывает следующий

Постулат 1. С каждой изолированной физической системой связывается комплексное векторное пространство со скалярным произведением, которое называется *пространством состояний* системы. Система полностью описывается *вектором состояния*, который представляет собой единичный вектор в пространстве состояний системы.

Изменения со временем состояния $|\psi\rangle$ квантово-механической системы описываются следующим постулатом.

Постулат 2. Эволюция замкнутой квантовой системы описывается унитарным преобразованием. Другими словами, состояние $|\psi\rangle$ системы в момент времени t_1 связано с её состоянием $|\psi'\rangle$ в момент времени t_2 посредством унитарного оператора U, зависящего только от моментов времени t_1 и t_2: $|\psi'\rangle = U|\psi\rangle$.

Пусть $|\psi\rangle = |x_1, x_2, \ldots, x_k\rangle$ и $|\psi'\rangle = |y_1, y_2, \ldots, y_k\rangle$. Равенство $|\psi'\rangle = U|\psi\rangle$ в обозначениях квантовых схем представлено на рис. 1.

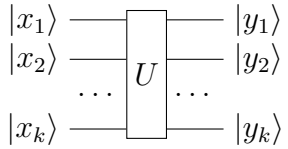
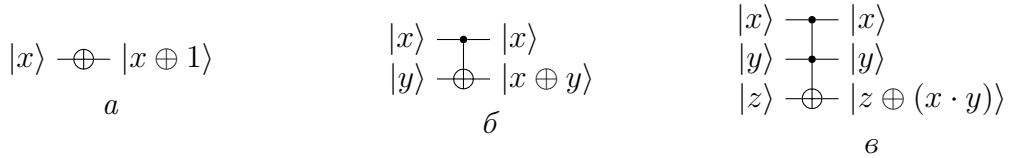


Рис. 1. Квантовая схема равенства $|\psi'\rangle = U|\psi\rangle$

Базовое преобразование квантового компьютера будем называть *вентилем*. В настоящей работе для построения всех операций и функций на квантовом компьютере используются базисные вентили, представленные на рис. 2 ($x, y, z \in \mathbb{F}_2$).



а *б* *в*

Рис. 2. Используемые вентили: *а* — вентиль Паули-X (NOT); *б* — вентиль CNOT; *в* — вентиль Тоффоли (CCNOT)

Определим *глубину квантовой схемы* [4] как количество слоёв, которые содержит схема. Один слой состоит из базисных вентилей, применённых к непересекающимся множествам кубит.

3. Модель оракула с квантовым списком

Рассмотрим предложенную в [5] модель оракула, при которой список L хранится в квантовой памяти. Работа модели происходит следующим образом:

1) получение номера вектора на вход и передача его в переключатель;
2) выбор вектора из списка по выходу переключателя и копирование его;
3) проверка скопированного вектора на условие поиска;
4) вывод ответа: 1 — если вектор удовлетворяет условию, 0 — если нет.

В данной работе для построения квантового оракула рассматривается подход, минимизирующий количество кубит. *Постоянными* будем называть те кубиты, которые используются на протяжении всей работы оракула, а *временными* — те, которые нужны только во время проведения операции. В табл. 1 приведены количество кубит и глубина схемы в реализации операций, используемых в построении оракула.

Таблица 1

Количество кубит и глубина схемы, используемые в предложенной реализации, без учёта длины входа

Операция	Кубиты		Глубина
	постоянные	временные	
Сложение двух целых m-битных чисел, представленных в дополнительном коде	$m+1$	–	$3m$
Возведение в квадрат целого m-битного числа, представленного в прямом коде	$2m-2$	m^2-2m	$10m^2-33m+22$
Переключатель, где номер вектора представляется целым m-битным числом	2^m	–	3^m-1
Получение отрицания целого m-битного числа, представленного в дополнительном коде	m	–	$m+2$
Перевод целого m-битного числа из дополнительного кода в прямой	m	$m-2$	$2\lceil\log_2(m-1)\rceil+m+2$
Сравнение двух целых положительных m-битных чисел	1	$m+1$	$7m$
Функция $F:\mathbb{F}_2^n\to\mathbb{F}_2^m$	m	$n+\left\lceil\frac{m}{4}\right\rceil-3$	$2^{n+1}-2n+\lceil\log_2 m\rceil(2^{n+1}-3)-3$

С помощью данных, представленных в табл. 1, получается следующая

Теорема 1. Пусть имеется список длины K, состоящий из целочисленных векторов размерности $d\geqslant 2$, каждая координата которых кодируется битовой строкой длины $m\geqslant 3$. Тогда для реализации квантового оракула, при котором список L хранится в квантовой памяти, потребуется не более

$$\lceil\log_2 K\rceil+2^{\lceil\log_2 K\rceil}+Kdm+3dm^2+13dm+2d+6m+3\lceil\log_2 d\rceil+3$$

кубит. При этом глубина не превосходит

$$2 \cdot 3^{\lceil \log_2 K \rceil} + 2K(2\lceil \log_2 dm \rceil + 1) + 20m^2 + 4\lceil \log_2 m \rceil + \\ + \lceil \log_2 d \rceil (3\lceil \log_2 d \rceil + 12m + 5) + 21.$$

Как видно из теоремы 1, хранение списка L в квантовой памяти приводит к линейному росту используемого числа кубит от длины K, которая растёт экспоненциально с увеличением размерности решётки.

4. Модель оракула с классическим списком

Рассмотрим другую модель оракула, при которой список L хранится в классической памяти. Работа модели происходит следующим образом:

1) получение номера вектора на вход и передача его в функцию F;
2) получение в качестве выхода функции F вектора из списка L, соответствующего заданному номеру;
3) проверка скопированного вектора на условие поиска;
4) вывод ответа: 1 — если вектор удовлетворяет условию, 0 — если нет.

Данная модель помогает избежать линейного роста количества кубит, используемых оракулом, от увеличения размера списка L. Однако хранение списка L и построение векторной булевой функции F потребуют на стороне классической части памяти и вычислений, которые линейно зависят от длины списка L.

Теорема 2. Пусть имеется список длины K, состоящий из целочисленных векторов размерности $d \geqslant 2$, каждая координата которых кодируется битовой строкой длины $m \geqslant 3$. Тогда для реализации квантового оракула, при котором неупорядоченный список L хранится в классической памяти, потребуется не более

$$\lceil \log_2 K \rceil + 13dm + 5d + 6m + 3\lceil \log_2 d \rceil + 3 + \max\left(3d(m^2-1), \lceil \log_2 K \rceil + \left\lceil \frac{dm}{4} \right\rceil - 3\right)$$

кубит. При этом глубина не превосходит

$$2^{\lceil \log_2 K \rceil + 2} - 4\lceil \log_2 K \rceil + \lceil \log_2 dm \rceil (2^{\lceil \log_2 K \rceil + 2} - 6) + 20m^2 + \\ + \lceil \log_2 d \rceil (3\lceil \log_2 d \rceil + 12m + 5) + 4\lceil \log_2 m \rceil + 17.$$

Как видно из теоремы 2, число кубит, используемое для реализации квантового оракула с классическим списком, растёт логарифмически от длины списка L.

5. Связь числа кубит и глубины схемы квантового оракула с параметрами криптосистем

В табл. 2 приведены количество кубит и глубина схемы, достаточные для реализации квантовых оракулов с различными видами списка при атаке на постквантовые криптосистемы.

NTRU [6], SABER [7] и CRYSTALS-Kyber [8] являются постквантовыми криптосистемами, основанными на решётках и прошедшими в финальный раунд конкурса NIST. Как видно из табл. 2, экспоненциальная длина списка L накладывает ограничения на возможность реализации гибридных атак на полноразмерные постквантовые криптосистемы. Однако стоит заметить, что для атак на NTRU используются циклические решётки, исследование которых может помочь уменьшить длину списка L, что приведёт к меньшим верхним оценкам на сложность реализации квантового оракула.

Таблица 2

Вид списка	Уровень защищ.	NTRU		SABER		CRYSTALS-Kyber	
		Кубиты	Глубина	Кубиты	Глубина	Кубиты	Глубина
Квантовый	1	$2^{227,48}$	$2^{340,19}$	$2^{228,95}$	$2^{343,36}$	$2^{228,85}$	$2^{343,36}$
	3	$2^{298,45}$	$2^{452,72}$	$2^{337,06}$	$2^{512,95}$	$2^{336,96}$	$2^{512,95}$
	5	$2^{359,31}$	$2^{547,82}$	$2^{444,99}$	$2^{684,12}$	$2^{444,89}$	$2^{684,12}$
Классический	1	$2^{19,4}$	$2^{219,91}$	$2^{19,77}$	$2^{221,91}$	$2^{19,6}$	$2^{221,91}$
	3	$2^{19,81}$	2^{291}	$2^{20,36}$	2^{329}	$2^{20,18}$	2^{329}
	5	$2^{20,28}$	2^{351}	$2^{20,77}$	2^{437}	$2^{20,6}$	2^{437}

Заключение

Получены новые уточнённые верхние оценки на сложность реализации квантового оракула, использующего квантовый список L, из алгоритма Гровера для реализации гибридного квантово-классического алгоритма на основе GaussSieve, который может быть использован для атак на криптосистемы, стойкость которых зависит от решения задачи SVP. Предложена новая модель оракула с классическим списком L, на сложность реализации которой также получены верхние оценки. Проанализирована сложность реализации рассмотренных моделей оракула для атаки на постквантовые криптосистемы, основанные на решётках и являющиеся финалистами конкурса NIST.

ЛИТЕРАТУРА

1. *Laarhoven T., Mosca M., and van de Pol J.* Finding shortest lattice vectors faster using quantum search // Des. Codes Cryptogr. 2015. V. 77. No. 2. P. 375–400.
2. *Micciancio D. and Voulgaris P.* Faster exponential time algorithms for the shortest vector problem // Proc. 21 Ann. ACM-SIAM Symp. Discrete Algorithms. Society for Industrial and Applied Mathematics, USA, 2010. P. 1468–1480.
3. *Grover L. K.* A fast quantum mechanical algorithm for database search // Proc. 28 Ann. ACM Symp. Theory of Computing. N.Y.: ACM, 1996. P. 212–219.
4. *Nielsen M. A. and Chuang I. L.* Quantum Computation and Quantum Information. Cambridge: Cambridge University Press, 2010.
5. *Бахарев А. О.* Разработка и анализ оракула для гибридной атаки на криптографическую систему NTRU с использованием алгоритма квантового поиска // Прикладная дискретная математика. Приложение. 2021. № 14. С. 62–67.
6. *Chen C., Danba O., Hoffstein J., et al.* NTRU Algorithm Specifications and Supporting Documentation. `https://ntru.org/f/ntru-20190330.pdf`.
7. *D'Anvers J.-P., Karmakar A., Roy S. S., and Vercauteren F.* SABER: Mod-LWR based KEM. `https://www.esat.kuleuven.be/cosic/pqcrypto/saber/files/saberspecround1.pdf`.
8. *Avanzi R., Bos J., Ducas L., et al.* CRYSTALS-Kyber Algorithm Specifications and Supporting Documentation. `https://cryptojedi.org/papers/kybernist-20171130.pdf`.