УДК 621.391

ЗАЩИТА КАНАЛОВ ПЕРЕДАЧИ И ХРАНЕНИЯ ДАННЫХ НА ОСНОВЕ АЛГЕБРАИЧЕСКИХ РЕШЕТЧАТЫХ КОДОВ

М.А. АЛИСЕЕНКО, С.Б. САЛОМАТИН

Белорусский государственный университет информатики и радиоэлектроники, Республика Беларусь

Поступила в редакцию 31 марта 2020

Аннотация. Рассмотрены алгебраические решетки, задачи нахождения кратчайшего и ближайшего векторов. Показан алгоритм приведения кратчайшего базиса решетки. Приведены достоинства и недостатки криптосистемы на решетках.

Ключевые слова: алгебраические решетки, CVP, SVP, LLL-алгоритм, NPA-алгоритм, NTRUEncrypt.

Введение

Поиск кратчайшего базиса решеток является базовой операцией для решения задач теории решеток. Эти задачи используются для создания схем стойкой криптографии, устойчивых к квантовым вычислениям. Процесс ортогонализации Грамма-Шмидта является частью алгоритма Ленстры-Ленстры-Ловаса (LLL) приведения базиса решетки. Для решения задачи поиска ближайшего вектора рассмотрен алгоритм Nearest Plane Algorithm (NPA). На решетках основывается криптосистема NTRUEncrypt, параметры которой должны быть взаимно простыми. Стойкость обеспечивается трудностью поиска кратчайшего вектора решетки.

Алгебраические решетки

Алгебраическая решетка является конечно порожденной аддитивной подгруппой множества R^n . Решетку L можно представить как множество целочисленных линейныхкомбинаций $L(b_1,\ldots,b_n) = a_1b_1 + \ldots a_nb_n$, $(a_1 \ldots a_n) \in Z^n$ n линейно независимых базисных векторов $\{b_1,\ldots,b_n\} \subset R^m$ в m -мерном евклидовом пространстве, где m и n – размерность и ранг решеткисоответственно. Решетки, у которых размерность m и ранг n равны, называются полноразмерными [1].Определитель решетки равен $\det L = \sqrt{\det(B^T B)}$, что равно объему фундаментального параллелотопа, образованного базисом $B = b_1,\ldots,b_n$, рис. 1.

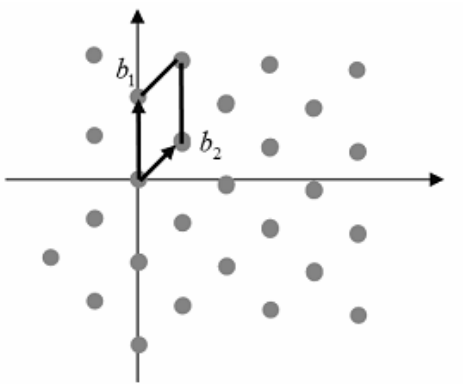

Рис. 1. Фундаментальный параллелотоп решетки, образованный базисом

Базис решетки не единственен: матрица перехода от одного базиса решетки к произвольному другому унимодулярна, т. е. ее определитель равен ±1, поэтому детерминант решетки не зависит от выбора базиса [2]. Произведение базисной и унимодулярной матрицы даст новый базис решетки.

Некоторые задачи теории решеток используется для создания схем стойкой криптографии, которые устойчивы для квантовых вычислений. Задача нахождения кратчайшего вектора (SVP, Shortest Vector Problem) подразумевает нахождение в заданном базисе решётки ненулевой вектор по отношению к определённой нормали. Математическая запись кратчайшего вектора:

$$a^* = \arg.\min_{a \in Z^n \setminus \{0\}} \|Aa\|^2 = \arg.\min_{a \in Z^n \setminus \{0\}} a^T G a, \quad (1)$$

где A – полноранговая матрица, являющаяся базисом решетки, $G = A^T A$ – матрица Грамма решетки.

Задача нахождения ближайшего вектора (CVP, Closest Vector Problem) – нахождение вектора в решётке по заданному базису и некоторому вектору, не принадлежащему решетке, при этом максимально схожего по длине с заданным вектором.Математическая запись ближайшего вектора к произвольному вектору y :

$$a^* = \arg.\min_{a \in Z^n} \|Aa - y\|^2 = \arg.\min_{a \in Z^n} (a^T G a - 2y^T Aa + y^T y). \quad (2)$$

По аналогии с линейными кодами решетка может быть выражена через порождающую матрицу и целочисленный коэффициент, что показано в выражении:

$$\Lambda = \{\lambda = \underbrace{[b_1 \vdots \cdots \vdots b_n]}_{G} a : a \in Z^n\}. \quad (3)$$

Под кратчайшим вектором решетки понимается вектор, длина которого:

$$\lambda(\Lambda) = \min_{x,y \in \Lambda, x \neq y} \|x - y\| = \min_{x \in \Lambda, x \neq 0} \|x\|. \quad (4)$$

Первый последовательный минимум, под которым понимается наименьший радиус окружности (шара), соответствует длине кратчайшего вектора решетки, рис. 2.

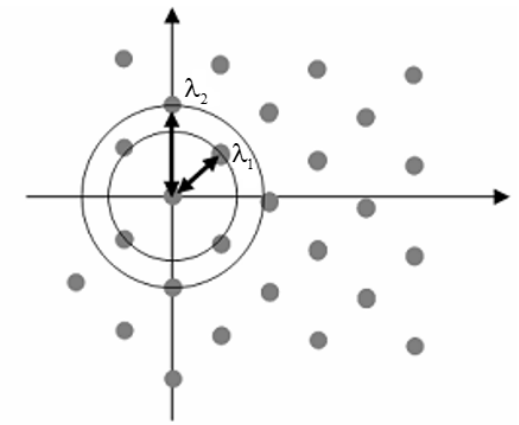

Рис. 2. Кратчайший базис-радиус решетки

Задачи теории решёток можно решить, если базис решетки редуцирован, т.е. состоит из относительно коротких почти ортогональных векторов. На сегодняшний день эффективным алгоритмом редукции базиса решетки является алгоритм LLL (Ленстры-Ленстры-Ловаса).

Алгоритм Ленстры-Ленстры-Ловаса (LLL)

За полиномиальное время алгоритм преобразует базис на решетке в кратчайший почти ортогональный базис на этой же решетке.Для векторного пространства R^n процесс Грамма-Шмидта позволяет преобразовать произвольный базис в ортонормированный («идеал», к которому стремится LLL-алгоритм), но не гарантирует того, что на выходе каждый из векторов будет целочисленной линейной комбинацией исходного базиса. Таким образом, полученный в результате набор векторов может и не являться базисом исходной решётки. Это привело к необходимости создания специального алгоритма для работы с решетками[3]. Процесс ортогонализации базиса описывается выражением:

$$b_1^* = b_1, b_n^* = b_n - \sum_{k=1}^{n-1} \frac{(b_n, b_k^*)}{(b_k^*, b_k^*)} b_k^*. \quad (5)$$

Для каждого *n*существует такое число $c(n)$, что в любой *n*-мерной решетке L можно выбрать базис $b_1,\ldots,b_n$, для которого произведение норм векторов меньше или равно произведению числа $c(n)$ и определителя решетки $\|b_1\| \cdot \ldots \cdot \|b_n\| \le c(n) \cdot \det(L)$. В алгоритме LLLконстанта $c(n) = 2^{n(n-1)/4}$ [4].

В начале работы алгоритма задан базис решетки $b_1,\ldots,b_n$ и параметр (норма) δ. Норма зависит от контекста задачи; обычно используются нормы $l_1 = \max_j \sum_i |a_{ij}|$, Евклида, Чебышева.На выходе алгоритма – приведенный LLL-базис.

1. Если какой-либо коэффициент по модулю больше $1/2$, то из вектора b_k необходимо вычесть вектор b_j, домноженный на округленный коэффициент.

2. Пересчет коэффициента $\mu_{k,j} = \frac{(b_n, b_k^*)}{(b_k^*, b_k^*)}$ для $j < k$.

3. Проверка условия $\left|b_k^* + \mu_{k,k-1} b_{k-1}^*\right| \ge \frac{3}{4}\left|b_{k-1}^*\right|^2$. Если оно не выполняется, то индексы проверяемых векторов меняются местами. Условие проверяется для вектора уже с новым индексом.

4. Пересчет $b_k^*, b_{k-1}^*, \langle b_k^*, b_k^* \rangle, \langle b_{k-1}^*, b_{k-1}^* \rangle$ для $k > 1$ и $\mu_{k-1,j}, \mu_{k,j}$ для $j < k$.

5. Возврат к шагу 1, если остался коэффициент $\mu_{k,j}$,по модулю превышающий $1/2$.

Если Λ – решетка в Z^n с базисом $b_1,\ldots,b_n$, причем $|b_i| \le B, i = 1,\ldots,n$,где, $B \in R, B \ge 2$, то алгоритм построения LLL-приведенного базиса делает $O(n^4 \log B)$ арифметических операций. При этом целые числа, встречающиеся в ходе работы алгоритма, имеют двоичную длину $O(n \log B)$ битов.

Алгоритм NPA

Алгоритм для решения проблемы ближайшего вектораCVP (Closest Vector Problem) известен под названием Nearest Plane Algorithm (NPA), разработан L. Babai. Коэффициент аппроксимации алгоритма составляет $2\left(\frac{2}{\sqrt{3}}\right)^n$, где n – это ранг решетки. Во многих приложениях алгоритм применяется с неизменным *n*, что дает постоянный коэффициент приближения. Аппроксимацию CVPопределяют как проблему поиска (search), оптимизации (optimization) и решения проблемы (decision)gap. Установивкоэффициентприближения $\gamma = 1$ [5]:

– (CVP_γ, search) – при заданном базисе $B \in Z^{m \times n}$ и точке $t \in Z^m$ найти точку $x \in L(b)$ такую, что $\forall y \in L(b), \|x - t\| \le \gamma \|y - t\|$;

– (CVP_γ , optimization)– при заданном базисе $B \in Z^{m\times n}$ и точке $t \in Z^m$ найти $r \in Q$ такую, что $dist(t, L(B)) \le r \le \gamma \cdot dist(t, L(b))$;

– (CVP_γ ,decision)– при заданном базисе $B \in Z^{m\times n}$ и точке $t \in Z^m$ и $r \in Q$ решить $dist(t, L(B)) \le r$ или $dist(t, L(B)) \ge \gamma \cdot r$.

Алгоритм содержит два этапа: сначала применяется редукция LLLc $\delta = 3/4$ для входной решетки, затем идет поиск целочисленной комбинации базисных векторов $B \in Z^{m\times n}$, которая близка к искомому вектору $t \in Z^m$. Этот шаг аналогичен внутреннему циклу на этапе редукции алгоритма LLL. На выходе алгоритма вектор $x \in L(b)$ такой, что $\|x - t\| \le 2^{\frac{n}{2}} dist(t, L(B))$.

Представление второго этапа алгоритма следующее. Если ортонормированный набор неполной решетки задан $\bar{b}_1 / \|\bar{b}_1\|, \ldots, \bar{b}_n / \|\bar{b}_n\|$, то необходимо расширить базис на $m - n$ векторов. Используя такой базис, матрица B B и вектор t следующие:

$$\begin{pmatrix} \|\bar{b}_1\| & * & \cdots & * \\ \vdots & \|\bar{b}_2\| & \cdots & * \\ & & \ddots & \vdots \\ & & & * \\ & & & \|\bar{b}_n\| \\ 0 & \cdots & & 0 \\ 0 & \cdots & & 0 \\ \vdots & \cdots & & \vdots \\ 0 & \cdots & & 0 \end{pmatrix} \begin{pmatrix} * \\ * \\ \vdots \\ * \\ * \\ * \\ * \\ \vdots \\ * \end{pmatrix}. \qquad (6)$$

Алгоритм ищет целочисленную комбинацию столбцов, для которых каждая координата $i = 1, \ldots, n$ находится в пределах $\pm 1/2 \|\bar{b}_i\|$ из i-ой координаты t. Таким образом, алгоритм сначала находит кратный n -ыйстолбец матрицы, который дает n -ю координату с точностью до $\pm 1/2 \|\bar{b}_i\|$, затем он продолжает до $n-1$ столбца и т.д. Если решетка не является полгораноговой, то последние $m - n$ величины соответствуют пространству, ортогональному размерности решетки (span).

Геометрическое описание второго этапа представлено на рис. 3.

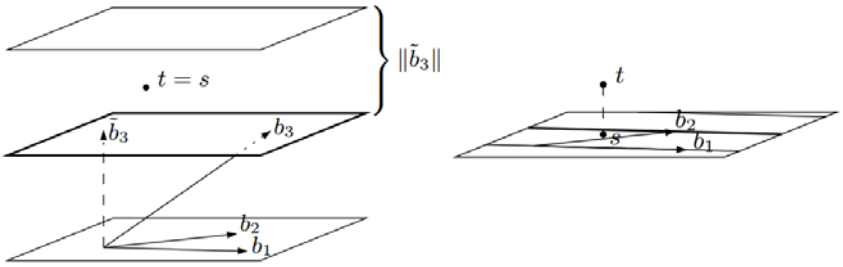

Рис. 3. Алгоритм NPA для решетки ранга 3

1. Пусть s – это проекция t на $span(b_1, \ldots, b_n)$.

2. Нахождение c такой, что гиперплоскость $c\bar{b}_n + span(b_1, \ldots, b_{n-1})$ максимально приближена к s.

3. Пусть $s' = s - cb_n$. Сдвиг s' и $L(b_1, \ldots, b_{n-1})$, пусть искомое – это x'.

4.Возврат $x = x' + cb_n$.

Таким образом, x является ближайшим вектором к s в $cb_n + L(b_1, \ldots, b_{n-1})$.

Криптосистема NTRUEncrypt

Криптосистема NTRUEncrypt, основанная на решетчатой криптосистеме, создана в качестве альтернативы RSA и криптосистемам на эллиптических кривых (ECC). Стойкость алгоритма обеспечивается трудностью поиска кратчайшего вектора решетки, которая более стойкая к атакам, осуществляемым на квантовых компьютерах. Алгоритм использует операции над кольцом $Z[X]/(X^N-1)$ усеченных многочленов степени, не превосходящей $N-1$:

$$a(X)=a=a_0+a_1X+\cdots+a_{N-1}X^{N-1},\ a_0,\ldots,a_{N-1}\in Z. \quad (7)$$

Такой многочлен можно также представить вектором

$$\vec{a}(X)=\vec{a}=\sum_{i=0}^{N-1}a_iX^i=[a_0,\ldots,a_{N-1}]. \quad (8)$$

Криптосистема определяется рядом параметров: N, p, q. Для сохранения стойкости алгоритма необходимо, чтобы параметры p и q были взаимно простыми, их набольший общий делитель должен равняться единице.

Для передачи сообщения необходимо сгенерировать открытый и закрытый ключи. случайным образом выбирает два малых полинома f и g из кольца усеченных многочленов R. Для определения малости полиномов используются числа d_f и d_g которые выбирает отправитель. Малость полиномов означает, что относительно произвольного полинома по модулю q, в котором коэффициенты равномерно распределены, у малого полинома они будут много меньше q [6].

Полином f имеет d_f коэффициентов, равных 1, (d_f-1) коэффициентов, равных -1, и остальные, равные 0. Полином g будет иметь d_g коэффициентов, равных 1, d_g равных -1, и остальные, равные 0. Выбранные полиномы должны держаться в тайне, т.к. они используются для расшифровки сообщения.

Далее вычисляются обратные полиномы f_p и f_q по модулю p и q соответственно, такие, что $f\times f_p=1(\mathrm{mod}\,p)$ и $f\times f_p=1(\mathrm{mod}\,q)$. Если эти обратные полиномы не существуют, то полином f выбирается заново. Секретный ключ – это пара (f, f_p), а открытый ключ h вычисляется по формуле $h=p\times f_q\times g(\mathrm{mod}\,q)$.

Для отправки сообщения с помощью открытого ключа h необходимо представить сообщение в виде полинома m с коэффициентами по модулю p, выбранными из диапазона $(-p/2, p/2]$. Затем необходимо выбрать другой малый полином r, который является ослепляющим, и вычислить шифротекст $e=(r\times h+m)(\mathrm{mod}\,q)$.

Для расшифровки полученного сообщения е используется секретный ключ и вычисляется $a=f\times e(\mathrm{mod}\,q)$.

Коэффициенты выбираются из диапазона $(-q/2, q/2]$, затем вычисляется $b=a(\mathrm{mod}\,p)$.

Используя вторую часть секретного ключа, исходное сообщение $c=f_з\times b(\mathrm{mod}\,p)$.

Открытый ключ так же может быть представлен эрмитовой матрицей $2n\times 2n$:

$$B=\begin{pmatrix} qI_n & 0 \\ H & I_n \end{pmatrix}. \quad (9)$$

где I_n единичная матрица $n\times n$, q – целое число (обычно $q=2^8$ или 2^{10}), а H – матрица $n\times n$ с элементами в $\{0,1,\ldots q-1\}$.Матрица H строится таким образом, чтобы базис решетки, порожденной B, состоял из коротких векторов[3]. Матрица H является циклической матрицей, т.е. строки матрицы H – это циклические повороты первого ряда матрицы H. Это означает, что для задания открытого ключа необходимо указать только q и первую строку матрицы H; открытый ключ требует $O(n\log_2 q)$ бит.

Решетчатая конструкция NTRU позволяет криптосистеме противостоять атакам «квантового типа». При максимальных настройках безопасности NTRU на четыре порядка быстрее RSA и на три порядка быстрее ECC [7].

Затратной операцией алгоритма является операция умножения элементов кольца. Ее решением может стать использование китайской теоремы об остатках (CRT) для замены операций умножения на операции сложения при небольших значениях параметра p.

Для хранения полиномов f и g необходимо $2N$ бит, а для полинома h необходимо $N \log_2 q$ бит [8].

Из преимуществ NTRUEncrypt перед аналогом – криптосистемой RSA можно указать более высокую скорость работы. Выполнение операций шифрования и дешифрования требует $O(n^2)$ операций, в отличие от $O(n^3)$ у того же RSA. Недостаток системы – необходимость использования рекомендованных параметров.

Заключение

Алгоритмы на основе теории решеток позволяют обеспечивать стойкость криптографических систем, использующих вычисления на решетках. Необходимо решать задачи кратчайших, ближайших векторов и поиска кратчайшего базиса решетки. Криптосистема NTRUEncrypt, построенная на решетках, быстрее, чем RSAи ECC, однако для заданных уровней стойкости требует использования рекомендованных параметров.

PROTECTION OF DATA TRANSMISSION AND STORAGE CHANNELS BASED ON ALGEBRAIC LATTICE CODES

M.A. ALISEYENKA,S.B. SALOMATIN

Abstract. Algebraic lattices and problems of finding the shortest and closest vectors are considered. An algorithm for reducing the shortest lattice basis is shown. The advantages and disadvantages
of the cryptosystem on lattices are given.

Keywords: lattice, CVP, SVP, LLL-algorithm, NPA-algorithm, NTRUEncrypt.

Списоклитературы

1. КузьминО.В., Усатюк В.С. // Программные продукты и системы №4. 2012. С.180–183.
2. Пискова А.В., Менщиков А.А., Коробейников А.Г. // Вопросы о кибербезопасности №1(14). 2016. С.47–52.
3. GalbraithS. // MathematicsofPublicKeyCryptography.2012. P. 365–381.
4. Шокуров А.В., Кузюрин Н.Н., Фомин С.А. Решетки, алгоритмы и современная криптография : учеб. пособие.М.: Институт системного программирования РАН. 2011.
5. RegevO., KaplanE. LatticesinComputerScience. TelAvivUniversity, 2004.
6. Разумов Е.В. // Компьютерные системы и сети : материалы 49-й научной конференции аспирантов, магистрантов и студентов. (Минск, 6 – 10 мая 2013 года). Минск: БГУИР, 2013. С. 61–62.
7. Червоная О.В., Сушко С.А. Криптоалгоритм NTRU, ГВУЗ «Национальный горный университет», 2011.
8. HoffsteinJ., [et. al.] Practicallattice-basedcryptography: NTRUEncrypt and NTRUSign.M., 2005.