Федеральное государственное бюджетное образовательное учреждение высшего образования
МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ имени М.В.ЛОМОНОСОВА

Факультет вычислительной математики и кибернетики
Кафедра информационной безопасности

На правах рукописи

ЧЕРЕПНЕВ Михаил Алексеевич

О вычислительной сложности алгоритмов факторизации и дискретного логарифмирования

01.01.09 - Дискретная математика и математическая кибернетика

ДИССЕРТАЦИЯ
на соискание ученой степени
доктора физико-математических наук

Научный консультант:
академик РАН Соколов И.А.

Москва - 2017

Оглавление

I Введение **1**

II Положения, выносимые на защиту **8**

III Основное содержание диссертации **40**

1 Показательные сравнения **41**

1.1 О подъёме решений показательных сравнений 41

1.2 Сравнительная стойкость протокола Диффи-Хеллмэна и задачи дискретного логарифмирования 46

1.3 О полиномиальной эквивалентности задач дискретного логарифмирования и Диффи-Хеллмэна на некоторых стандартных кривых 56

2 Некоторые свойства распределения простых чисел **60**

3 Использование некоммутативной операции для построения систем защиты информации **71**

4 Задача Бризолиса **91**

5 Гиперэллиптические кривые **101**

5.1 Арифметика дивизоров 101

5.1.1 Алгоритмы сложения и приведения дивизоров 102

5.1.2 Алгоритмы приведения матриц 113

5.2 Ядро спуска Вейля 122

5.2.1 Исследование группового гомоморфизма метода спуска Вейля 122

6 Решение разреженных систем линейных уравнений над $GF(2)$ **132**

6.1 Замечания о симметрических матрицах 134
6.2 Блочные алгоритмы 139
6.2.1 Новый алгоритм типа Ланцоша-Паде (первая версия) 139
Вероятностный анализ 165
6.2.2 Распараллеленные версии блочных алгоритмов с распределёнными операциями 170
6.2.3 Итоговые оценки и тесты 185
6.2.4 Дальнейшие исследования 187
Новый алгоритм типа Видемана-Копперсмита 187
Связь между приближениями рядов по положительным и отрицательным степеням формальной переменной 192

7 Решение разреженных систем линейных уравнений над $GF(p)$ 209
7.1 A-ортогональный базис пространства Крылова и метод Ланцоша 210
7.2 A-ортогональный базис пространства Крылова и матричные приближения Паде 213
7.3 Рекуррентные формулы для приближений Паде 216
7.4 Блочный метод Ланцоша-Паде 219
7.5 Универсальный блочный метод Ланцоша-Паде 229

IV Заключение 244

V Приложение 248

1 Основные определения и теоремы в методе спуска Вейля 249

2 Алгоритм Ланцоша-Монтгомери 264

3 Алгоритм Видемана-Копперсмита 271

4 Реализация последовательной версии нового алгоритма решения разреженных систем линейных уравнений 272

Часть I

Введение

Диссертация состоит из введения, 7 глав и приложения. Все результаты, приведённые в основной части диссертации (1-7 глава), являются новыми, снабжены полными доказательствами и разъяснениями используемых понятий и определений.

С основами теории чисел можно познакомиться по изданиям [2, 3, 4, 5, 66, 7].

Основная часть настоящей работы посвящена хорошо известным функциям: дискретному логарифмированию (1,2,5 и 7 глава) и факторизации (3,6 глава). 1 глава посвящена оценкам вычислительной сложности задач дискретного логарифмирования и Диффи-Хеллмэна. Во 2 главе анализируется вычислительная сложность атакующих алгоритмов на схему электронной подписи стандарта ГОСТ Р 34.10-2012. В 3 главе исследована вычислительная сложность задачи фактоизации, возникающей при анализе схемы В.М.Сидельникова открытого распределения ключей. В главе 4 рассмотрены свойства неподвижных точек в задаче дискретного логарифмирования. В 5 главе исследовано ядро спуска Вейля, применённого к задаче дискретного логарифмирования на эллиптических кривых над полем характеристики 2. В 6 главе предлагается несколько модификаций нового алгоритма решения больших разреженных систем линейных уравнений над полем из двух элементов. В 7 главе эта техника распространяется на случай большого простого поля.

Схема открытого распределения ключей Диффи-Хеллмэна широко используется в современных программных продуктах, например, $Windows Server 2003 - 2008$. Задача её вскрытия очевидно не сложнее задачи дискретного логарифмирования, которая на сегодняшний момент в общем случае считается достаточно сложной.

В первой главе сложность алгоритма решения задачи дискретного логарифмирования в общем случае оценивается сверху через сложность алгоритма решения задачи Диффи-Хеллмэна. Полученная теорема в ряде случаев приводит к полиномиальной эквивалентности этих задач. Полученная

оценка обобщает результаты работы Боера 1988 года [91], и работы Маурера 1994 года [94] на общий случай. Данная теорема вошла в обзор [26] в качестве одного из итоговых результатов. В первой главе вводится функция максимальной длины дерева Пратта $s(m)$ числа m. А именно: для произвольного $m = \prod_{i=1}^{r} {p_i}^{\alpha_i}$ рассмотрим разложение на простые множители чисел $p_i - 1, i = 1, ..., r$, затем разложения $q_{ij} - 1$ для простых $q_{ij} \mid p_i$ и так далее. Получившееся разветвление называется деревом Пратта числа m, а встречающиеся простые числа его узлами. Это дерево впервые рассмотрел в 1975 году Пратт [24] и предложил его для использования в алгоритме проверки простоты. Считая расстояние между соседними узлами за единицу, обозначим $s = s(m)$ длину наибольшей ветви дерева m. Такая функция натурального аргумента рассмотрена автором впервые. Показано, как при помощи этой функции выразить сложность задачи дискретного логарифмирования через сложность задачи Диффи-Хеллмэна. В 1997 году Шуф [25] показал, что универсальный алгоритм (”generic algorithm”), решающий задачу Диффи-Хеллмэна, не может быть слишком простым (имеет экспоненциальную сложность для групп простого порядка). Это приближает полученную в первой главе верхнюю оценку для задачи дискретного логарифмирования к тривиальной. Однако основная теорема первой главы применима не только к универсальным алгоритмам, но и к алгоритмам, работающим с элементами фиксированной группы, связанными несколькими специальными групповыми операциями. Кроме того, эта теорема даёт конструктивный алгоритм решения задачи дискретного логарифмирования из алгоритма решения задачи Диффи-Хеллмэна. Есть перспектива построения групп, порядок которых имеет ограниченную длину ветвей дерева Пратта, в которых задачи дискретного логарифмирования и Диффи-Хеллмэна, являясь полиномиально эквивалентными, являются также и достаточно сложными.

Функция $s(m)$ была в 2008 году исследована Фордом, Конягиным и Люка [28], которые получили для неё нетривиальные оценки. Полученная ими оценка

сверху отличается от тривиальной несколько лучшей константой в показателе. Поэтому её применение совместно с основной теоремой первой главы не даёт новой информации о связи сложностей задач дискретного логарифмирования и Диффи-Хеллмэна. В этой же работе была высказана гипотеза об асимптотическом поведении функции $s(m)$ как $O(\ln \ln m)$ для почти всех m. Применение этой оценки совместно с теоремой первой главы уже приводит к нетривиальным результатам. А именно, в случае полиномиальности решения задачи Диффи-Хеллмэна получается алгоритм решения задачи дискретного логарифмирования, работающий "почти" полиномиально (лучше субэкспоненты).

Предложенный метод применён к оценке сложности алгоритма решения задачи дискретного логарифмирования в группе точек эллиптической кривой. Показано, что для некоторых эллиптических кривых, удовлетворяющих действующему стандарту ГОСТ Р 34.10-2012, алгоритмы решения задач дискретного логарифмирования и Диффи-Хеллмэна полиномиально эквивалентны. Более того, для них есть конструктивные полиномиальные алгоритмы решения одной при помощи оракула другой.

В первой главе доказаны теоремы о величине и числе простых делителей чисел вида $p - 1$, где p - простое число. Эти теоремы предоставляют информацию о структуре одного шага дерева Пратта, которую можно использовать и при построении алгоритма дискретного логарифмирования с оракулом Диффи-Хеллмэна. Используемые соображения позволили как следствие получить новую оценку снизу функции Эйлера для почти всех аргументов.

Во второй главе при помощи спаривания Эйта построен полиномиальный алгоритм сведения задачи Диффи-Хеллмэна на эллиптической кривой к решению некоторой системы полиномиальных уравнений. Получена оценка на степень этой системы.

Задача дискретного логарифмирования исследуется достаточно давно. Возникает естественный вопрос: можно ли использовать для реализации

схемы Диффи-Хеллмэна вместо дискретного логарифмирования в конечной группе другую функцию. В.М.Сидельников в совместной работе с автором диссертации [35] предложил использовать функцию разложения на множители в мультипликативной группе с некоммутативной операцией. Для стойкости соответствующей схемы распределения ключей эта функция должна быть трудновычислимой. Однако конкретных примеров таких групп пока нет. В третьей главе диссертации проведён анализ некоторых известных некоммутативных групп. Получены полиномиалные алгоритмы сведения задачи разложения на множители в них к решению некоторых систем линейных уравнений. Далее предложен пример некоммутативной мультипликативной операции с большей сложностью задачи разложения на множители.

В четвёртой главе было изучено сравнение

$$x \equiv R^x (mod p), x, R \in \mathbb{Z}, (R, p) = 1.$$

В современной теоретико-числовой литературе такие решения носят название неподвижных точек в задаче дискретного логарифмирования. Автору диссертации удалось построить аналитические формулы для вычисления пар (x, R), удовлетворяющих рассматриваемому сравнению, отличных от общеизвестных: $(g, g^{g^{-1}(mod(p-1))})$, где g - первообразный корень, удовлетворяющий условию Бризолиса: $(g, p-1) = 1$. В этих формулах R - уже необязательно первообразный корень. Доказана теорема о существовании решения с R, имеющим достаточно большой простой порядок. Получены оценки числа первообразных корней, удовлетворяющих условию Бризолиса, дающие оценку сложности алгоритма их построения.

Пятая глава посвящена анализу алгоритма спуска Вейля решения задачи дискретного логарифмирования на эллиптических кривых над полем характеристики 2. Основным инструментом в этом исследовании выбрано представление Мамфорда приведённых дивизоров парой многочленов. Данное

представление можно корректно связать с классами дивизоров (по модулю дивизоров рациональных функций). Метод спуска Вейля применяется к задаче дискретного логарифмирования в группе классов дивизоров. Доказано, что при выполнении некоторого достаточно легко проверяемого условия порядок ядра преобразования спуска Вейля не делится на 2. Это показывает, что в случае когда порядок группы классов дивизоров делится на 2, это преобразование нетривиально.

В шестой главе предложен новый алгоритм решения разреженных систем линейных уравнений над GF(2), которые возникают при решении задачи целочисленной факторизации алгоритмами с факторной базой. Показано, что параллельная реализация построенного алгоритма имеет асимптотически лучшую оценку времени работы, чем известные оценки для алгоритма Видемана-Копперсмита, который был применён при постановке последнего рекорда факторизации чисел RSA [162].

В седьмой главе результаты шестой главы перенесены на случай большого конечного поля. Полученный алгоритм может быть быть взят за основу при решении задачи дискретного логарифмирования при помощи факторной базы.

Предложенная в последних двух главах техника не только сохраняет возможность частичного распараллеливания высокого уровня при решении таких систем (то есть часть алгоритма может быть реализована на не связанных друг с другом кластерах), но и позволяет вообще исключить использование кластеров большой мощности. Другими словами, позволяет распараллелить весь алгоритм на связанные медленным каналом ($Internet$) вычислители, единственное требование к которым - это возможность содержать в оперативной памяти рассматриваемую задачу (матрицу линейной системы). Помимо уменьшения времени решения линейной системы, предложенный алгоритм не требует кратного увеличения оперативной памяти сверх объёма, необходимого для хранения указанной матрицы, как это было при постановке рекорда целочисленной

факторизации в декабре 2009 года [162].

Часть II

Положения, выносимые на защиту

1. В общем случае получена оценка сверху на сложность задачи дискретного логарифмирования через сложность задачи Диффи-Хеллмэна. Данная оценка в ряде случаев приводит к полиномиальной эквивалентности этих задач. Полученная оценка обобщает результаты работы Боера 1988 года и работы Маурера 1994 года на общий случай. Эта теорема вошла в обзор Маурера и Вулфа 2001 года "Diffie-Hellman Protocol" в качестве одного из итоговых результатов. Впервые рассмотрена функция максимальной длины дерева Пратта $s(m)$ числа m. Это дерево введено в рассмотрение в 1975 году Праттом, который предложил его для использования в алгоритме проверки простоты. В 1997 году Шуф показал, что алгоритм общего применения (generic algorithm), решающий задачу Диффи-Хеллмэна, не может быть "простым" (имеет экспоненциальную сложность для групп простого порядка). Однако, доказанная теорема об оценке применима не только к алгоритмам общего применения, но и к алгоритмам, работающим с элементами фиксированной группы, связанными несколькими специальными групповыми операциями.

Функция $s(m)$ была в 2008 году исследована Фордом, Конягиным и Люка, которые получили для неё нетривиальные оценки. Полученная ими оценка сверху отличается от тривиальной несколько лучшей константой в показателе. Поэтому её применение совместно с рассматриваемой оценкой не даёт новой информации о связи сложностей задач дискретного логарифмирования и Диффи-Хеллмэна. В этой же работе была высказана гипотеза об асимптотическом поведении функции $s(m)$ как $O(\ln \ln m)$ для почти всех m. Применение этой оценки уже приводит к нетривиальным результатам. А именно, в случае полиномиальности решения задачи Диффи-Хеллмэна получается алгоритм решения задачи дискретного логарифмирования, работающий "почти" полиномиально (лучше субэкспоненты).

Изложенный метод, применён к оценке сложности задачи дискретного логарифмирования на группе точек эллиптической кривой простого порядка

p, где $p-1$ раскладывается на маленькие взаимнопростые множители. Такой случай возможен для кривых, удовлетворяющих ГОСТ Р 34.10-2012. Показано, что в этом случае задачи дискретного логарифмирования и Диффи-Хеллмэна полиномиально эквивалентны. Кроме того, построен конструктивный полиномиальный алгоритм вычисления дискретного логарифма при помощи оракула Диффи-Хеллмэна.

2. Доказаны теоремы о величине и числе простых делителей чисел вида $p-1$, где p - простое число. Эти теоремы предоставляют информацию о структуре одного шага дерева Пратта, которую можно использовать и при построении алгоритма дискретного логарифмирования с оракулом Диффи-Хеллмэна. Получена новая оценка снизу функции Эйлера для почти всех аргументов.

3. Проведён анализ некоторых известных некоммутативных групп на стойкость построенной на их основе схемы открытого распределения ключа, предложенной В.М.Сидельниковым [35]. Доказано, что задача разложения на множители в них является полиномиальной, а схема В.М.Сидельникова с их использованием — нестойкая. Предложен пример некоммутативной мультипликативной операции с несколько большей сложностью задачи разложения на множители.

4. Показано, что задача универсальной подделки ГОСТ Р 34.10-1994 [19] сводится к нахождению неподвижных точек в задаче дискретного логарифмирования. Построены аналитические формулы для вычисления пар (x,R), удовлетворяющих сравнению

$$x \equiv R^x((\mathrm{mod}p), x, R \in \mathbb{Z}, (R,p)=1,$$

отличные от общеизвестных: $(g, g^{g^{-1}((\mathrm{mod}p-1)}))$, где g - первообразный корень, удовлетворяющий условию $(g,p-1)=1$ (условие Бризолиса). В этих формулах R уже необязательно первообразный корень. Доказана теорема о существовании решения с R, имеющим достаточно большой простой порядок. Получены оценки

на число первообразных корней, удовлетворяющих условию Бризолиса.

5. Проведён анализ метода спуска Вейля при решении задачи дискретного логарифмирования на эллиптических кривых над полем характеристики 2. Основным инструментом в этом исследовании выбрано представление Мамфорда приведённых дивизоров парой многочленов. Данное представление можно корректно связать с классами дивизоров (по модулю дивизоров рациональных функций).

Доказано, что при выполнении некоторого достаточно легко проверяемого условия порядок ядра преобразования спуска Вейля не делится на 2. Это означает, что в случае, когда порядок группы точек на эллиптической кривой делится на 2, сужение преобразования спуска Вейля на эту группу нетривиально.

6. Проанализирована сложность линейного этапа алгоритма факторизации целых чисел. Предложен новый алгоритм решения разреженных систем линейных уравнений над $GF(2)$, которые возникают при решении этой задачи алгоритмами с факторной базой. Доказано, что параллельная реализация построенного алгоритма работает быстрее алгоритма Видемана-Копперсмита 1993г., который был применён при постановке последнего рекорда факторизации чисел RSA в декабре 2009 г. Предложенная техника не только сохраняет возможность частичного распараллеливания высокого уровня при решении таких систем (то есть часть алгоритма может быть реализована на не связанных друг с другом кластерах), но позволяет вообще исключить использование кластеров большой мощности. Другими словами, эта техника позволяет распараллелить весь алгоритм на связанные медленным каналом ($Internet$) вычислители, единственное требование к которым - это возможность содержать в оперативной памяти рассматриваемую задачу (матрицу линейной системы). Помимо уменьшения времени на решение линейной системы, предложенный алгоритм не требует кратного увеличения оперативной памяти сверх объёма, необходимого для хранения указанной матрицы, как это

было при постановке рекорда целой факторизации в декабре 2009 года при помощи алгоритма Видемана-Копперсмита. Таким образом, предложенный алгоритм превосходит алгоритм Видемана-Копперсмита по всем основным параметрам. Преимуществом по сравнению с алгоритмом Монтгомери 1994г. является более высокая точка насыщения, что позволяет решать значительно большие системы на многопроцессорной вычислительной технике. Таким образом, предложенный алгоритм оказывается на сегодняшний день наилучшим для использования на многопроцессорной технике с несколькими сотнями независимых вычислительных узлов.

7. Построен новый алгоритм решения разреженных систем линейных уравнений над $GF(p)$, которые возникают при решении задачи дискретного логарифмирования алгоритмами с факторной базой. Этот алгоритм обладает теми же преимуществами, что и, описанный выше, алгоритм для поля $GF(2)$.

Краткое содержание диссертации.

В первой главе диссертации рассматриваются алгоритмы дискретного логарифмирования и связанные с их применением вопросы распределения целых простых чисел. Задача дискретного логарифмирования рассматривается в конечных полях, кольцах вычетов, на эллиптических кривых, а также в общей постановке. Соответствующие определения можно посмотреть в [10, 11, 12, 13, 14, 15, 16, 17].

В некоторых случаях дискретное логарифмирование оказывается простой задачей и имеет даже аналитическое выражение [18].

Частным Ферма называется функция

$$Q_m \pmod m : \quad (\mathbb{Z}/m^2\mathbb{Z})^* \to \mathbb{Z}/m\mathbb{Z},$$

определяемая по модулю m, где

$$Q_m(a) = \frac{a^{L(m)}-1}{m},$$

где $L(m)$ - функция Кармайкла, максимальный порядок элементов

мультипликативной группы вычетов по модулю m (универсальная экспонента) [18].

Далее получены некоторые обобщения результатов работы [18] (см. [19, 20]), а именно, получена формула для подъёма решений в случае, когда основание дискретного логарифма по простому модулю не является первообразным корнем.

Пусть $p \mid Q_p(g)$. Обозначим $l = \gamma_p(Q_p(g)) \in \mathbb{N}$, то есть $p^l \parallel Q_p(g)$ (или $p^l \mid Q_p(g), p^{l+1} \nmid Q_p(g)$).

Теорема 1 ([19, 20, 21]). *Пусть p – простое число, $(g,p) = 1, l = \gamma_p(Q_p(g)) \in \mathbb{N}, \alpha \in \mathbb{N} \setminus \{1\}$. Тогда, если сравнение*

$$a \equiv g^x \pmod{p^\alpha} \tag{1}$$

разрешимо, то его решение x удовлетворяет следующим условиям

1. *При $\alpha \in \{1, \ldots, l\}$ выполнено $ord_{p^\alpha} g = ord_p g$ и x есть единственное* $\pmod{ord_p g}$ *решение сравнения*

$$a \equiv g^x \pmod{p} \tag{2}$$

2. *При $\alpha \geq l+1, k = \max\{l+1, \alpha - (l+1)\}$ выполнено равенство $ord_{p^\alpha} g = p^{\alpha-(l+1)} ord_p g$ и x есть единственное* $\pmod{p^{\alpha-(l+1)} ord_p g}$ *решение системы*

$$\begin{cases} a \equiv g^x \pmod{p} \\ x\frac{Q_{p^{k-l}}(g)}{p^l} \equiv \frac{Q_{p^{k-l}}(a)}{p^l} \pmod{p^{\alpha-(l+1)}}. \end{cases} \tag{3}$$

Данная теорема даёт формулы с меньшим модулем и меньшей степенью, чем в случае перехода к основанию, являющемуся первообразным корнем [18].

Наиболее распространённой схемой распределения ключа, основанной на задаче дискретного логарифмирования, является схема Диффи-Хеллмэна с использованием односторонней функции возведения в степень в конечной группе. Она, наряду со схемой RSA, была положена в основу широко применяемой схемы " Kerberos " .

Для вскрытия схемы Диффи-Хеллмэна необходимо решить следующую задачу: по известным p, g, g^a, g^b найти g^{ab}. Эта задача носит название задачи Диффи-Хеллмэна. В диссертации вопрос о полиномиальной эквивалентности алгоритмов решения задачи Диффи-Хеллмэна и задачи дискретного логарифмирования для произвольной конечной мультипликативной группы сводится [23] к оценкам длины максимальной ветви дерева Пратта [24] порядка этой группы.

Пусть $G(t, m)$ — произвольная циклическая группа порядка m с операцией, которую будем в дальнейшем обозначать $+$, требующей для своего выполнения t битовых операций, и пусть $L(t, m)$ обозначает минимальное количество битовых операций, необходимых для решения задачи дискретного логарифмирования в группе $G(t, m)$, то есть при известных $a, b \in G(t, m)$ найти $n \in Z_m$ такое, что

$$a = nb, \tag{4}$$

здесь nb есть $b + \cdots + b$, где элемент b повторяется n раз.

Будем предполагать, что решение уравнения (4) существует.

Не ограничивая общности дальнейших рассуждений, будем считать b образующим группы $G(t, m)$.

Пусть $D(t, m)$ — не убывающая по m оценка количества битовых операций, необходимых для решения задачи Диффи-Хеллмэна в $G(t, m)$: при известных a_1, a_2 и b, таких, что $a_1 = n_1 b, a_2 = n_2 b$, найти

$$a_3 = (n_1 n_2) b. \tag{5}$$

Пусть $D^*(t, m)$ — также неубывающая по m оценка количества битовых операций, необходимых для решения той же задачи при помощи алгоритмов, количество битовых операций в которых удовлетворяет неравенству

$$D^*(t, m) \leq t D^*(C, m),$$

для некоторой абсолютной константы C (например, таких, которые используют только операции, совокупная сложность которых не более, чем совокупная сложность используемых групповых операций).

Для произвольного $m = \prod_{i=1}^{r} {p_i}^{\alpha_i}$ рассмотрим разложения на простые множители чисел $p_i - 1, \quad i = 1, \ldots, r$, разложения $q_{ij} - 1$ для простых $q_{ij} \mid p_i$ и так далее. Получившееся разветвление называется деревом Пратта [24] числа m, а встречающиеся простые числа - его узлами. Узлы q_{ij} свяжем с узлом p_i. Назовём ветвью максимальную последовательность связанных друг с другом узлов. Обозначим $s = s(m)$ длину наибольшей ветви дерева m.

Назовём задачу дискретного логарифмирования в группе $G(t, m)$ сертифицированной, если заранее известны все простые числа, стоящие в узлах дерева Пратта числа m,а также хотя бы один первообразный корень для каждого такого простого числа.

Теорема 2 ([23]). *Для сертифицированной задачи дискретного логарифмирования*

$$L(t, m) \leq s \log^2 m D(\ldots D(D(t, m), m) \ldots), \tag{6}$$

где справа изображена s- кратная композиция функции $D(t, m)$ по первому аргументу, а логарифм берётся по некоторому постоянному основанию, не зависящему от t, m.

$$L(t, m) \leq ts \log^2 m (D^*(C, m))^s. \tag{7}$$

Данная теорема обобщает результаты работы [94] на случай $s > 1$ для негладких чисел. Функция $s(m)$ была введена и рассмотрена впервые.

Отметим, что после публикации этой теоремы в работе [25] было показано, что любой вероятностный алгоритм, решающий задачу Диффи-Хеллмэна в произвольной группе фиксированного порядка с вероятностью не меньше константы ("generic algorithm"), не может иметь сложность меньше $O(\sqrt{p})$, где

p - максимальный простой делитель порядка. В теореме 2 исходная группа фиксирована, а её результат применим для алгоритмов, работающих лишь в исходной группе и группах, состоящих из её элементов с операциями специального вида, получаемых из исходной и связанными с оракулом Диффи-Хеллмэна.

Теорема 2 вошла в обзор [26].

В 2008 году, K. Ford, С.В.Конягин, F.Luca [28] доказали, что для почти всех натуральных чисел m выполнено $c_1 \log_2 \log_2 m \leq s(m) \leq c_2 (\log_2 m)^{1-0.0378}$ с некоторыми абсолютными константами c_1, c_2. Там же высказана гипотеза о том, что точная верхняя оценка совпадает с приведённой здесь нижней. В этом случае предыдущая теорема даёт "почти" полиномиальную зависимость между сложностями алгоритмов решения задач дискретного логарифмирования и Диффи-Хеллмэна.

Для группы точек на эллиптической кривой хорошо известны эндоморфизмы $[n]-n$-кратного сложения точек с собой. Пусть используемая циклическая группа точек $< P >$ кривой имеет простой порядок p. Пусть $p-1 = \prod_{i=1}^{t} {q_i}^{\alpha_i}$ - разложение $p-1$ на простые множители. В стандарте ГОСТ Р 34.10-2012 не указано никаких требований на это разложение.

Пусть по двум точкам эллиптической кривой над простым полем из r элементов $Q, P \in E[\mathbb{F}_r]$, связанным равенством

$$Q = [n]P,$$

надо найти n, определённое по модулю p. Обозначим сложность алгоритма решения этой задачи $DLE(r,p)$, а сложность алгоритма вычисления $[n_1 n_2]P$ по паре $([n_1]P, [n_2]P)$ обозначим $DHE(r,p)$. Будем полагать, что этот последний алгоритм не проще одной операции на эллиптической кривой $E[\mathbb{F}_r]$. Символом log будем обозначать логарифм по некоторому фиксированному основанию, значение которого каждый раз будет некоторой эффективной абсолютной константой.

Теорема 3 ([163, 164, 82]). $DLE(r,p) \leq \log p DHE(r,p) \sum_{i=1}^{t} \pi {q_i}^{\alpha_i}$.

В случае если ${q_i}^{\alpha_i}$ для любого i невелики, получаем полиномиальную эквивалентность алгоритмов решения задач дискретного логарифмирования и Диффи-Хеллмэна. Построенный алгоритм с оракулом, а также некоторые его модификации, описанные в первой главе диссертации, применены в ОАО "Концерн "Автоматика" к анализу систем защиты информации. Представленные нижние оценки для сложности задачи дискретного логарифмирования использованы в качестве одного из оснований для увеличения мощности используемого простого поля, о чем имеется акт внедрения.

Для оценки параметра $s(m)$ необходимо изучить распределение простых делителей чисел вида $p-1$ для простых p. Некоторые свойства этого распределения приведены во второй главе.

Пусть $N(M)$ — количество элементов в множестве M, $v(n)$ - количество различных простых делителей числа n, $v_{>t}(n)$ - количество различных простых делителей числа n, больших t.

Теорема 4 ([30]). *Для любого положительного* ε

$$N(p \leq x | \text{для любого простого} \quad q : p-1 = qn, n \in \mathbb{N}, q > e^{lnx/lnlnx} \text{выполнено}$$

$$v(q-1) \in [(1-\varepsilon)lnln\frac{x}{n}, (1+\varepsilon)lnln\frac{x}{n}]) = \frac{x}{lnx} + \overline{o}(\frac{x}{lnx}).$$

Теорема 5 ([29]). *Для любого* $\varepsilon \in (0, \frac{1}{3})$ *и* $t = e^{(lnx)^{\frac{1-\varepsilon}{2e}}}$

$$N(p \leq x | v(p-1) \in [(1-\varepsilon)lnlnx, (1+\varepsilon)lnlnx], v_{>t}(p-1) > \frac{1}{2}v(p-1)) = \frac{x}{lnx} + \overline{o}(\frac{x}{lnx}).$$

Ранее было известно лишь среднее значение функции $v(p-1)$ по простым p [93].

Кроме того, в диссертации получена новая оценка для функции Эйлера для почти всех натуральных чисел.

Теорема 6 ([31]). *Для некоторой абсолютной константы c справедливо равенство*

$$N\left(n \le x | \varphi(n) > \frac{cn}{\ln\ln\ln n}\right) = x + \overline{o}(x)$$

Приведённый анализ может быть усилен применением спаривания Эйта [82].

В третьей главе исследуется идея построения протокола открытого распределения ключей на основе некоммутативной операции, выдвинутая В.М. Сидельниковым. Представлен анализ стойкости этого протокола при использовании некоторых естественных некоммутативных операций [35]. В частности, построены алгоритмы сведения задачи вскрытия этого протокола к системам линейных уравнений.

Предположим, что в открытом доступе имеется описание некоторой группы G и двух ее коммутативных подгрупп H и R, при этом предполагается, что не единичные элементы $h \in H, r \in R$ не коммутируют. Абоненты A и B для выработки общего секретного ключа поступают следующим образом. Каждый из них независимо друг от друга случайно вырабатывает по одному элементу из H и R и в последующем держит их в секрете. Значком $\in_R$ будем обозначать случайный выбор.

A		**B**
$h_A \in_R H, r_A \in_R R$		$h_B \in_R H, r_B \in_R R$
$g_A = h_A r_A$		
	$\xrightarrow{g_A}$	
		$g_B = h_B r_B$
	$\xleftarrow{g_B}$	
$h_A g_B r_A$	$=$	$h_B g_A r_B$

Из определений вытекает, что

$$h_A g_B r_A = h_A(h_B r_B)r_A = (h_A h_B)(r_B r_A) = (h_B h_A)(r_A r_B) =$$

$$h_B(h_A r_A)r_B = h_B g_A r_B = g$$

Тем самым, общий секретный ключ g выработан.

Теорема 7 ([35]). *Описанный выше протокол В.М.Сидельникова нестоек (предложены полиномиальные алгоритмы нахождения общего секретного ключа по открытой информации) для циклических подгрупп мультипликативной группы матриц, образов эллиптических модулей в кольце эндоморфизмов конечного поля, а также их факторов по степени автоморфизма Фробениуса.*

Статья [35] написана в соавторстве. Сама идея построения протокола распределения ключей на основе некоммутативной операции, предложена В.М.Сидельниковым. Идея использования одной некоммутативной полугруппы вместо двух принадлежит В.В.Ященко. Указанная выше теорема доказана автором диссертации.

Далее в диссертации построен некоторый более стойкий пример [36, 37] ассоциативной операции для протокола В.М.Сидельникова на основе обобщённых символов Лежандра и Якоби. Приведены примеры, когда эта операция быстро вычисляется.

Для использования построенной операции интересен следующий пример быстровычислимой логарифмической функции. Рассмотрим целые алгебраические числа в простом круговом поле $\mathbb{Q}(\xi), \xi = e^{\frac{2\pi i}{p}}$, p - простое. Пусть $\lambda = 1 - \xi$. Рассмотрим мультипликативно независимые в $\mathbb{Z}[\xi]$ элементы $\eta_i = 1 - \lambda^i, p - 1 \geq i \geq 1$. Известно, что эти элементы вместе с некоторой степенью первообразного корня по модулю p образуют мультипликативный базис кольца $\mathbb{Z}[\xi]$, и для любого $z \in \mathbb{Z}$ имеем

$$1 - z\lambda \equiv {\eta_1}^{e_1(z)} \cdot \dots \cdot {\eta_{p-1}}^{e_{p-1}(z)} (mod \lambda^p),$$

где $0 \leq e_i(z) \leq p-1$.

Гипотеза.(опубликована в [38]) Пусть $z \in \mathbb{Z}$, тогда $e_n(z)$ является многочленом от z, в который z входит только в степенях, являющихся делителями n. В случае простого $n = q$ верна формула:

$$e_q(z) \equiv \frac{1}{q}(z^q - z)(mod p).$$

Эта высказанная мной гипотеза недавно была доказана Раинчик В.С. [39].

Если в первой главе диссертации анализировалось использование задачи дискретного логарифмирования в схемах открытого распределения ключей, то в четвёртой главе [19, 20] проанализировано использование этой задачи в схемах шифрования и электронной подписи Эль-Гамаля. Впервые показано, что задача универсальной подделки подписи ГОСТ Р34.10-1994 сводится к решению сравнения

$$r \equiv R^r \pmod{p}. \tag{8}$$

Умение решать это сравнение относительно $r \in \{1, \dots, p-1\}$ даст возможность подписывать любое сообщение, то есть осуществлять универсальную подделку. Отметим, что решение рассматриваемого сравнения не сводится сразу к задаче дискретного логарифмирования. Аналогичные рассмотрения для схем цифровой подписи Эль-Гамаля и DSA приводят к сравнению вида $r \equiv CR^r \pmod{p}, r \in \{1, \dots, p-1\}$, что снова приводит к сравнению (8) с условием $r = C_1 r', r' \in \{1, \dots, p-1\}$. С помощью китайской теоремы об остатках рассматриваемые сравнения могут быть легко решены для $r \in \{1, \dots, p(p-1)\}$.

$$\begin{cases} r \equiv c_1 \pmod{p-1} \\ r \equiv CR^{c_1} \pmod{p} \end{cases} \tag{9}$$

для произвольного $c_1 \pmod{p(p-1)}$. Однако, алгоритма решения сравнения (8) относительно $r \in \{1, \ldots, p-1\}$ пока не построено. В диссертации рассмотрены алгоритмы построения решений этого сравнения относительно пары $x, R \in \{1, \ldots, p-1\}$.

Получены некоторые оценки на число первообразных корней, удовлетворяющих условию Бризолиса:

$$(g, p-1) = 1. \tag{10}$$

Эти первообразные корни дают решение задачи Бризолиса

$$x \equiv R^x \pmod{p}; x, R \in \{1, \ldots, p-1\}, R - \text{первообразный корень} \pmod{p} \tag{11}$$

в виде $x = g, R \equiv g^{g^{-1} \pmod{p-1}} \pmod{p}$.

Теорема 8 ([45]). *Пусть p, d - простые, $d \mid p-1, d \geq \sqrt{p}+1$. Тогда существует (x, R) - решение уравнения (11) такое, что $ord_p R = d$.*

Теорема 9 ([45]). *Пусть $p = 2^s q + 1 > 2^{2^s}$, где p - простое, и выполнено одно из двух условий:*

1) q - простое нечётное, или

2) q - любое нечётное, $q \mid ord_p 2$.

Тогда сравнение (11) выполнено при

$$x = 2^{2^s}, R \equiv g^{qt + ind_g 2(\frac{q+1}{2})^{2^s - s}} \pmod{p}$$

для любого g - первообразного корня по модулю p, и $t \in \mathbb{N}$. Кроме того, для t, при которых показатель в последнем сравнении нечётный, R будет первообразным корнем по модулю p.

Теорема 10 ([45]). *Пусть $p \geq 5$ простое, тогда случайное нечётное число $g \in \{1, \ldots, p-1\}$ будет первообразным корнем по модулю p, удовлетворяющим условию (10) с вероятностью не меньше:*

1) $\frac{3\varphi(p-1)}{p-1} - 1$ *при* $p \equiv 1 \pmod 4$*;*
2) $\frac{4\varphi(p-1)}{p-1} - \frac{7}{4} - \frac{1}{2(p-1)}$ *при* $p \equiv -1 \pmod 4$.

Некоторые близкие к последней теореме формулы получены Кобели и Захареску (1999). Однако указанная выше теорема из них не следует. В случае *1)* для чисел типа Софи Жермен она улучшает результат Захареску асимптотически. В остальных случаях даёт лучшую оценку для простых p с числом десятичных знаков, меньшим нескольких сотен (конкретное значение зависит от константы в O(), которая в работе Захареску не приведена).

В пятой главе рассматривается т.н. алгоритм "спуска Вейля" для задачи дискретного логарифмирования на эллиптических кривых. Об использовании эллиптических и гиперэллиптических кривых в системах защиты информации можно посмотреть в [50, 52, 53, 54, 55, 56, 57, 58]. В частности, получены более простые доказательства некоторых теорем о свойствах дивизоров неособой гиперэллиптической кривой [59]. Эти доказательства, в отличие от известных [60], используют представление Мамфорда дивизоров в виде пары многочленов.

Напомним, что дивизор, имеющий неотрицательные коэффициенты, называется приведённым, если его степень не превосходит рода кривой. Доказано следующее свойство представления Мамфорда:

Теорема 11 ([48])**.** *Если два приведённых дивизора эквивалентны (отличаются на дивизор рациональной функции), то представляющие их пары многочленов совпадают.*

Другими словами пары многочленов, степень которых ограничена родом кривой и удовлетворяющие условиям представления Мамфорда взаимнооднозначно соответствуют классам дивизоров.

Далее предложен новый алгоритм приведения матрицы с целыми рациональными коэффициентами к Смитовой нормальной форме [48, 90], необходимой при работе с дивизорами. Данный алгоритм обладает тем свойством,

что рост возникающих коэффициентов ограничен некоторой постоянной величиной, не зависящей, вообще говоря, от размера матрицы. Тем самым, он эффективнее алгоритма, предложенного для этих целей в известной книге Д.Кнута [49]. Одновременно с СНФ данный алгоритм вычисляет и матрицу приводящих к ней преобразований. В алгоритме Сторйоханна 2013 года [67] матрица преобразования строится после СНФ исходной матрицы с помощью нескольких дополнительных матричных умножений. Предложенный алгоритм строит матрицу преобразований параллельно с СНФ и имеет лучшие константы в главном члене оценки его сложности, чем алгоритм Сторйоханна.

Для решения задачи дискретного логарифмирования на эллиптической кривой над конечным полем характеристики 2 может быть применён спуск Вейля, а именно гомоморфизм исходной кривой на якобиан некоторой гиперэллиптической кривой (группа классов дивизоров) над меньшим по мощности полем. С основными свойствами и определениями можно познакомиться в [55, 61, 62, 63, 64].

Получен результат, описывающий ядро спуска Вейля [48, 51]. Метод спуска Вейля [9, 68, 69, 70] в случае поля K характеристики 2 заключается в построении гомоморфизма ϕ из группы точек $E(K)$ эллиптической кривой E

$$y^2 + xy + x^3 + \alpha x^2 + \beta = 0, \alpha, \beta \in K, \tag{12}$$

над большим полем $K = GF(2^{nr})$ в группу классов дивизоров некоторой гиперэллиптической кривой над относительно меньшим подполем $k = GF(2^r)$ поля K.

А именно, ϕ индуцировано композицией следующих замен переменных:

$$x = 1/f(u); \quad y = \sqrt{\beta} + \frac{v}{f^2(u)} \tag{13}$$

для некоторого многочлена $f(u) = \lambda_{-1} + \sum_{i=0}^{m-1} \lambda_i u^{2^i}, \lambda_i \in K; \lambda_0 \neq 0, \lambda_{m-1} \neq 0$. В новых координатах (см. [48, §2, гл.1, часть 1]) кривая E будет иметь вид

$$v^2 + f(u)v + h(u) = 0, \tag{14}$$

где $h(u) = f(u) + \alpha f(u)^2 + \sqrt{\beta} f(u)^3$.

Сделаем следующую замену переменных [70, (2.14), §2, гл.1, часть 1]:

$$\begin{cases} \widetilde{u} = \lambda_0 u + \lambda'' \\ \widetilde{v} = v + g(\widetilde{u})t(\widetilde{u}), \end{cases}$$

где $g(\widetilde{u}) = f(\frac{\widetilde{u}-\lambda''}{\lambda_0})$, а $\lambda'' \in K$ выбрано так, что $g(\widetilde{u}) \in k[\widetilde{u}]$,

$$t(\widetilde{u}) = Tr_{K/k}\left(\frac{v}{g(\widetilde{u})}\right) + \frac{v}{g(\widetilde{u})}.$$

В новых переменных кривая (14) запишется следующим уравнением над k:

$$\widetilde{v}^2 + g(\widetilde{u})\widetilde{v} + g(\widetilde{u})(1 + ag(\widetilde{u}) + bg(\widetilde{u})^2) = 0, \text{где } a, b \in k. \tag{15}$$

Теорема 12 ([48]). *Пусть n - нечётное число и род кривой (15) не равен $2^{m-1}-1$ (в этом случае он равен 2^m). Тогда для построенного методом спуска Вейля гомоморфизма ϕ выполнено:*

$P(0, \sqrt{\beta}) \notin Ker\phi$ и степени вхождения двойки в порядки точек кривой E и их образов совпадают.

С точки зрения практики, данная теорема подтверждает эффективность применения алгоритма спуска Вейля для решения задачи дискретного логарифмирования на эллиптических кривых над полем характеристики 2. Помимо упрощения арифметики, связанным с переходом в меньшее поле, мы можем применять на гиперэллиптической кривой алгоритм с факторной базой, который имеет субэкспоненциальную сложность. Таким образом, использование полей большой простой характеристики предпочтительнее. Результатов о характеристиках ядра спуска Вейля раньше не было.

Результаты 5 главы внедрены в 8 центре ФСБ РФ, имеется акт внедрения.

В шестой главе диссертации получен новый блочный алгоритм решения больших разреженных систем линейных уравнений над полем из двух элементов [72, 73, 75, 76, 77, 78, 79, 165]. В результате исследований получено несколько различных версий этого алгоритма. Показана эффективность предлагаемого нового метода по сравнению с известными алгоритмами Монтгомери и Видемана-Копперсмита.

Задача решения больших разреженных систем линейных уравнений над полем из двух элементов возникает, в частности, как этап в алгоритме факторизации целых рациональных чисел методом квадратичного решета или решета числового поля. Задача факторизации целых чисел возникает, в частности, при атаке на схему RSA.

Алгоритм Видемана-Копперсмита работает с произвольной исходной матрицей линейной системы. Алгоритм Монтгомери, также как и предлагаемый новый алгоритм, работает с симметричной матрицей исходной системы. Однако общий случай сводится к случаю симметричной матрицы.

Пусть $N, M \in \mathbb{N}$, и требуется найти нетривиальное решение системы линейных однородных уравнений вида

$$DX = 0, D \in \mathbb{F}^{M \times N}, X \in \mathbb{F}^{N \times n}, M < N, \tag{16}$$

где n – так называемый блочный параметр, обычно равный длине машинного слова, 32 или 64, или кратный этой длине.

Выберем случайно блок $Y \in \mathbb{F}^{N \times n}$ и рассмотрим систему

$$AX = B, A \in \mathbb{F}^{N \times N}, B, X \in \mathbb{F}^{N \times n}, \tag{17}$$

где $A = D^T D \in \mathbb{F}^{N \times N}$, в этих условиях вырожденная симметричная матрица. $B = AY$. Заметим, что любое решение X_D системы (16) позволяет построить X_A - решение системы (17) по формуле

$$X_A = X_D + Y.$$

Обратно, если есть X_A - решение системы (17), то $D^T D(X_A - Y) = 0$. Если $rangD = M$, то из полученного равенства находим, что $D(X_A - Y) = 0$. То есть, $X_A - Y$ - решение системы (16).

Обозначим $dim_{(16)}X$ размерность пространства, полученого в пересечении пространства $\langle X \rangle$, образованного столбцами произвольного блока X, с ядром линейного оператора D. Обозначим $dim_{(17)}X$ размерность пространства, полученого в пересечении пространства, образованного столбцами блока X, с ядром линейного оператора $A = D^T D$, а $dim'_{(17)}X$ - размерность пространства, полученого в пересечении пространства, образованного столбцами блока X, с ядром линейного оператора $A' = DD^T$.

Теорема 13. *[75] В приведённых обозначениях*

$$1) dim_{(16)}X \geq dim_{(17)}X - (M - rangD),$$

$$2) dim_{(16)}D^T X \geq dim'_{(17)}X - (M - rangD).$$

Пусть теперь требуется решить симметричную систему линейных уравнений:

$$Ax = B, A \in \mathbb{F}^{N \times N}, B \in \mathbb{F}^{N \times n}, \tag{18}$$

где A — симметричная матрица, а $\mathbb{F} = GF(2)$ — поле из двух элементов. Пусть n — длина машинного слова (32, 64 или 128), а d — оценка сверху на число ненулевых элементов в каждой строке матрицы A (плотность марицы).

Во всех рассматриваемых методах решения разреженных систем само решение ищется в пространстве Крылова $\langle B, AB, A^2B, \dots \rangle$. Предлагаемый алгоритм использует следующую теорему

Теорема 14 ([75]). *Пусть пространство Крылова $\langle W \rangle$ (здесь W – это конкатенация столбцов блоков $B, AB, A^2B, \dots$) построено в виде суммы*

непересекающихся A-ортогональных пространств $\langle W_i\rangle, i = 0, 1, \ldots, m$ с начальным блоком вида $W_0 = B = AY$, на первых m из которых A-скалярное произведение невырождено. Пусть $\langle W_m\rangle$ A-ортогонально всему $\langle W\rangle$. Пусть вычислено AW_m и X вида

$$X = \sum_{i=0}^{m-1} W_i (W_i{}^T A W_i)^{-1} W_i{}^T B, \tag{19}$$

$dim\langle W_m\rangle = \mu > 0$.

Тогда не более чем за $2\frac{(n+\mu-1)(n+\mu)}{2}N$ битовых операций с вероятностью не меньше $1 - \frac{1}{2^n}$ (при условии статистической независимости $\langle Y\rangle$ и $\langle W\rangle$) может быть вычислено решение системы (18).

Алгоритм Монтгомери, изложенный в [145], фактически, использует это утверждение без формулировки и доказательства.

Суть предлагаемого метода состоит в последовательном построении приближений Паде к некоторому ряду, коэффициенты которого зависят от матрицы линейной системы и блока векторов, стоящего в её правой части. Из этих приближений получаются блоки W_i, о которых шла речь в предыдущей теореме.

Пусть

$$\alpha(\lambda) = \sum_{i=0}^{\infty} \alpha_i \lambda^{-i}, \alpha_i \in \mathbb{F}^{n\times n}, \alpha_i = B^T A^i B$$

и для некоторых матричных многочленов $Q^{(i)}, P^{(i)}$ выполнено

$$\alpha(\lambda)\tilde{Q}^{(t)}(\lambda) + \tilde{P}^{(t)}(\lambda) = \sum_{i=t}^{\infty} \tilde{\rho}_i \lambda^{-i},$$

$$\alpha(\lambda)\tilde{Q}^{(t-1)}(\lambda) + \tilde{P}^{(t-1)}(\lambda) = \sum_{i=t-1}^{\infty} \tilde{\tau}_i \lambda^{-i},$$

$$\alpha(\lambda)Q^{(t-2)}(\lambda) + P^{(t-2)}(\lambda) = \sum_{i=t-1}^{\infty} \sigma_i \lambda^{-i}. \tag{20}$$

Теорема 15 ([74]). *Построен алгоритм, который строит приближения Паде* $Q^{(t)}(\lambda)$ *и приближения с порядком на единицу меньшим чем Паде* $\tilde{Q}^{(t)}(\lambda)$, *но с невырожденным старшим коэффициентом - единичной матрицей с вероятностью не меньше* $1 - \frac{1}{2^{n-1}}$.

Заметим, что вычисления по этой теореме требуют одновременного рекуррентного построения двух последовательностей приближений $Q^{(s)}(\lambda), \tilde{Q}^{(s)}(\lambda)$, в то время как по первой версии этой теоремы [72] строится только одна. Однако с точки зрения программирования эти вычисления эффективнее из-за фиксированного размера используемых матриц. Кроме того, здесь нет итераций, как в [72]. Поэтому число умножений на разреженную матрицу A для вычислений по формуле (21) становится равным $\frac{N}{n}$ — минимальному значению, необходимому для построения пространства Крылова.

Предложенный в [72] и в этой теореме алгоритм впервые дал рекуррентные формулы для построения приближений Паде к матричному ряду над поем из двух элементов. Отметим также, что эти теоремы применимы не только к кольцу матриц, но и к другим некоммутативным кольцам, в которых недоопределённая система имеет невырожденное решение.

Определим A-скалярное произведение двух матричных многочленов $(\varphi(\lambda), \psi(\lambda)) \in \mathbb{F}^{n\times n}$ как коэффициент при λ^{-1} в произведении $\varphi(\lambda)^T \alpha \psi(\lambda)$. Обозначим $(C, D) = C^T A D \in \mathbb{F}^{n\times n}$ матрицу попарных A-скалярных произведений вектор-столбцов матриц $C, D \in \mathbb{F}^{N\times n}$. Из этого определения и симметричности матрицы A следует, в частности, что $(C, D) = (D, C)^T$.

Далее из матричных приближений Паде строятся A-ортогональные блоки, образующие пространство Крылова. Это делается при помощи формулы:

$$Q^{(s)}(A,B) = \sum_{i=0}^{s} A^i B Q_i^{(s)} \tag{21}$$

с использованием следующей далее леммы

Лемма 1 ([71, 72]). *Пусть* $\alpha_i = B^T A^i B$, *тогда* $(\varphi(A,B), \psi(A,B)) = (\varphi(\lambda), \psi(\lambda))$.

Формула (21) является линейной по Q и B, а также удовлетворяет свойству

$$(\varphi(\lambda) \cdot \psi(\lambda))\,(A,B) = \psi(A, \varphi(A,B)). \tag{22}$$

Эта формула позволяет сократить до константы степень используемых в алгоритме многочленов, что приводит к существенной экономии вычислительных ресурсов.

Предложенный метод построения приближений Паде оптимизирован для удобства программирования.

Для получения линейной по размеру матрицы оценки сложности построения всех необходимых матричных приближений Паде предложена новая процедура. Суть её состоит в том, что после k-ого шага вычисление блочных векторов $A^i B, i = 0, 1, \ldots, k$ заменяется на вычисление блочных векторов вида

$$A^i Q^{(k)}(A,B), A^i Q^{(k-1)}(A,B), i = 0, 1, \ldots,$$

Очередные приближения Паде строятся как линейные комбинации приближений $Q^{(k)}(\lambda), Q^{(k-1)}(\lambda)$ с коэффициентами, являющимися многочленами ограниченной степени, откуда и получается линейный характер оценки сложности их построения. Необходимая для этого построения оперативная память даже меньше, чем для хранения исходной матрицы.

Основными операциями рассматриваемого алгоритма являются: умножение разреженной матрицы из $\mathbb{F}^{N\times N}$ на блочный вектор из $\mathbb{F}^{N\times n}$, умножение блочных векторов друг на друга (скалярное произведение), умножение блочных векторов на матрицы из $\mathbb{F}^{n\times n}$ (линейная комбинация).Рассмотрим две последние операции.

Согласно статье [145], сложность этих операций оценивается величиной $O(nN)$ операций с машинными словами. Однако для более быстрой реализации этих операций можно применить хорошо известный алгоритм "четырёх русских" [159], дающий, как мы покажем ниже, для умножения блока на маленькую матрицу существенно меньшее время.

Копперсмит для умножения блока на маленькую матрицу предложил фактически тот же алгоритм, что и алгоритм "четырёх русских", а для умножения блоков друг на друга - новый алгоритм.

Выбором оптимальных параметров в настоящей работе для этих алгоритмов получены следующие оценки

Лемма 2. *[76] Сложность вычисления скалярного произведения не более* $3\frac{nN}{log_2\frac{N}{2}-log_2log_2\frac{N}{2}}$, *а сложность вычисления линейной комбинации не более* $2\frac{nN}{log_2\frac{N}{2}}$.

Эти оценки приводят к тому, что рассматриваемые операции не влияют в общем случае на коэффициент при главном члене в оценке сложности всего алгоритма решения разреженной системы методом типа Ланцоша, хотя и близки при конкретных значениях параметров.

Получены результаты, учитывающие время на обмен между вычислительными узлами. Определим константу c следующим образом. Пусть время выполнения одной арифметической операции с машинными словами, умноженное на некоторую константу c, равно времени передачи одного машинного слова между вычислительными узлами. В современной компьютерной технике обмен между оперативной памятью и процессором осуществляется с $c \approx 5$, а между отдельными частями оперативной памяти: $c \approx 20$.

Теорема 16. *[78, 76] Оценка времени работы параллельной реализации с использованием блочного фактора алгоритма Видемана-Копперсмита с матричным умножением [148] (с построением базиса всех приближений "по дереву") при* $N \geq 2^{26}$ *и*

$$N > \frac{71log_2(32N)log_2(4N)}{cn}\left(\frac{240log_2(log_2(32N))log_2(\frac{60}{17n}log_2(log_2(32N)))}{17}\right)^4$$

имеет вид

$$94N^{1+\frac{3}{4}}n^{-\frac{1}{4}}c^{\frac{3}{4}}(log_2(32N)log_2(4N))^{\frac{1}{4}}.$$

С помощью алгоритма Видемана-Копперсмита, который был модифицирован Томэ, 12 декабря 2009 года был установлен рекорд целой факторизации чисел RSA. Асимптотическая оценка времени работы этой версии [161] с учётом распараллеливания умножения разреженной матрицы на блочный вектор:

$$O(N^{1+\frac{2}{3}}(logNloglogN)^{\frac{1}{3}}).$$

В реальных вычислениях 2009 года эта величина оказалась равной 120 суток. Пиковое использование оперативной памяти 1 ТВ.

Предлагаемые в диссертации алгоритмы построены с использованием блочных аппроксимаций Паде. Преимуществом этих алгоритмов по сравнению с алгоритмом Монтгомери 1995 года является более простая процедура вычисления скалярных произведений V^TAV без участия блоков из пространства Крылова. В алгоритме Монтгомери эту операцию труднее всего распараллелить. Её упрощение приводит к лучшим параллельным свойствам всего алгоритма и, в конце концов, к тому, что время работы сокращается практически до естественной границы — времени на построение пространства Крылова.

Проведённый анализ производительности соответствующих программ [76] показал лучшие параллельные свойства алгоритма, построенного на основе приближений Паде (оптимизированный алгоритм), по сравнению с существовавшими до этого программными реализациями других аналогичных алгоритмов. Это означает возможность эффективно использовать значительно больше вычислительных узлов.

Проведённый ИВМ РАН (с.н.с. Замарашкин Н.Л.) анализ производительности соответствующих программ [76] показал, что важным свойством предложенного подхода является то, что можно считать коэффициенты ряда на не связанных между собой вычислителях. Предложенный метод позволяет избежать роста необходимой оперативной памяти в процессе работы алгоритма. Это обстоятельство делает предлагаемый алгоритм принципиально лучше, чем алгоритм Видемана-Копперсмита и может позволить в будущем совсем отказаться от использования кластера при решении больших разреженных систем, ограничившись связанными в сети Интернет вычислителями, оперативная память которых может вместить матрицу исходной линейной системы. Связывающие их каналы должны допускать "синхронизацию", которая проводится 200 раз за весь алгоритм, при размере матрицы порядка $2 \cdot 10^8$. Современная динамика развития компьютерной техники показывает, что через 3-4 года таким требованиям будут удовлетворять персональные компьютеры, связанные сетью Интернет. Это значит, что для решения разреженных систем может быть привлечена вычислительная мощность, на несколько порядков превышающая мощность кластеров.

Предлагаемый алгоритм имеет дополнительный параметр (в тексте обозначен k), который позволяет управлять распараллеливанием. А именно, можно вручную задавать количество обменов между узлами, независимо вычисляющими свои части исходного ряда. Тем самым заключительный этап, требующий применения кластера с большой оперативной памятью, дробиться на k_1 этапов, обрабатывающих многочлены в k_1 раз меньшей степени. При этом k_1 раз приходится пересчитывать образующие. Оптимальное значение параметра k_1 выбирается в соответствии с пропускной способностью общей сети. Общее время работы программы будет подчиняться следующей зависимости:

$$k_1 L + \frac{c_1 N^2}{k_1},$$

где L - сложность решения задачи на кластере, c_1 - коэффициент, связанный

с пропускной способностью внешней сети. Арифметические вычисления показывают для матрицы порядка $2 \cdot 10^8$ и сети 1Гбит/сек время порядка 6 суток. Для сети 100Мбит/сек - 15 суток.

Для удобства читателя сгруппируем лучшие верхние оценки для разных алгоритмов при работе на одном кластере в таблицу

	Время
Wiedemann-Coppersmith with matrix polynomial multiplication	$94N^{1+\frac{3}{4}}n^{-\frac{1}{4}}c^{\frac{3}{4}}((log_2 32N)(log_2 4N))^{\frac{1}{4}}$
Wiedemann-Coppersmith-Thomé	$O(N^{1+\frac{2}{3}}(log_2 N log_2 log_2 N)^{\frac{1}{3}})$
Montgomery (1995) Алгоритм-K	$\frac{2c}{n}N^2$

Здесь n - длина машинного слова, а c — описанный выше коэффициент запаздывания передачи машинного слова по сравнению с производством одной операции с машинными словами. Число узлов выбрано так, чтобы минимизировать время работы соответствующего алгоритма. Таким образом показано, что оптимизированный алгоритм, предложенный автором, имеет асимптотически лучшую оценку сложности, чем алгоритм Видемана-Копперсмита, сохраняя при этом возможность распараллеливания высокого уровня, т.е. работу на не связанных друг с другом вычислительных узлах.

Эти оценки хорошо согласуются с результатами, полученными в ходе практической реализации нового алгоритма [76]. В рассмотренных диапазонах время сокращалось близко к обратно пропорциональной зависимости от числа

используемых вычислительных узлов, что является наилучшим показателем эффективности распараллеливания (см. [76] Таблицы 14, 15).

Приведём результаты численных экспериментов, проведённых в ИВМ РАН для матрицы «m512t» размером $5,5 \cdot 10^6$ и плотностью 51 на кластере Академии наук в одинаковых условиях.

Метод Монтгомери («m512t», МВС-1000, программа "Алгоритм-К"): 45 часов.

Первая версия предложенного метода («m512t», МВС-1000): 17 часов.

Реализованная первая версия была запрограммирована с отклонениями от оптимальной версии алгоритма для упрощения программирования. Показано, что программа, реализующая оптимальную версию нового алгоритма, будет работать примерно 8 часов столько же сколько и новейшая версия алгоритма Монтгомери с блочным фактором. При этом замеры времени показали, что ресурс использования параллельных вычислений для нового алгоритма выше, чем для алгоритма Монтгомери.

Суммируем преимущества предлагаемого алгоритма.

По сравнению с алгоритмом Монтгомери:

- Точка насыщения больше, чем для Монтгомери с блочным фактором. То есть есть возможность уменьшить время работы за счёт использования большего числа вычислительных узлов.
- Возможно распараллеливание высокого уровня (как в алгоритме Видемана-Копперсмита). Можно вести речь о решении в Интернете на вычислителях, память которых вмещает саму задачу.
- Линейные комбинации, скалярные произведения, умножения на разреженную

матрицу не чередуются, а группируются.

- Меньше линейных комбинаций в 1,5 раза.

По сравнению с алгоритмом Видемана-Копперсмита:

- Требуемая память не растёт в процессе работы (”синхронизация” приводит к тому, что степень приближений не растёт). Есть возможность исключить использование кластера.
- Трудоёмкость меньше (отсутствует логарифмический множитель в главном члене асимптотики).
- Проще анализ вероятности срабатывания.

Вот результаты замеров времени работы различных программ на 400 и 900 вычислительных узлах современных кластеров.

Алгоритм	400, МВС-1000 $(5,5 \cdot 10^6)$	400, Ломоносов $(33 \cdot 10^6)$	900, Ломоносов $(33 \cdot 10^6)$
P.Montgomery, Алгоритм-K	**45 часов**	-	-
P.Montgomery, J.Papadopulos	-	**80 часов**	-
P.Montgomery, J.Papadopulos, И.А.Поповян, Ю.В.Нестеренко	-	-	**54 часа**
М.А.Черепнёв, Н.Л.Замарашкин	**8 часов**	*9,4 часов*	*6,27 часа*

Наклонным шрифтом указано время, рассчитанное (эксперимент не проводился) исходя из того, что при фиксированном числе вычислительных узлов

время растет не быстрее, чем N^2, а увеличение в k раз числа используемых вычислительных узлов даёт уменьшение времени как минимум в $\sqrt{k}$ раз. Кроме того, узлы "Ломоносова" в 36,8 раз мощнее узлов МВС-1000 (99,5 Гфлопс против 2,7 Гфлопс), отношение соответствующих констант c примерно равно $1,8$, а $\sqrt{2,25} = 1,5$.

Отметим, что предложенный подход позволил сконструировать целую серию алгоритмов, которые имеют преимущества перед известными алгоритмами в случае использования вычислителя специфической структуры и мощности. Возможны и дальнейшие упрощения предложенной процедуры вычисления последовательных приближений Паде, что повлияет на сокращение времени работы всего алгоритма.

Кроме того, с теоретической точки зрения изложенный подход связывает между собой алгоритмы, основанные на ортогонализации пространства Крылова, и алгоритмы, основанные на поиске соотношения для рекуррентной последовательности. Давно замечено, что реализации этих подходов имеют практически одинаковую эффективность. Изложенный подход позволяет понять теоретические причины этого явления и использовать преимущества обоих методов.

Разработанный в диссертации алгоритм решения больших разреженных систем линейных однородных уравнений над $GF(2)$, который можно использовать для атаки на схему RSA, применён в ОАО "Концерн "Автоматика" при анализе систем защиты информации (получены новые параметры безопасности) о чем имеется акт внедрения.

В заключении шестой главы предложены две модификации алгоритма Видемана-Копперсмита, показывающие связь между этим алгоритмом и алгоритмом Монтгомери. Изменения коснулись той части алгоритма, где необходимо использовать кластер с большой оперативной памятью. Первым описан вариант с полной заменой техники построения матричных приближений,

предложенной Копперсмитом, на алгоритм построения приближений Паде для рядов по отрицательным степеням, разработанный автором диссертации [77].

Во втором варианте исходная техника Копперсмита дополнена анализом скалярных произведений получаемых приближений [80, 165]. Для получения решающего преимущества этот вариант необходимо дополнить процедурой "синхронизации". Построенный таким способом алгоритм будет примерно в 3 раза быстрее по сравнению с описанным в [79] оптимизированным алгоритмом. Это является следствием того, что построение приближений к рядам по положительным степеням несколько проще.

Идеи шестой главы могут быть применены и к обобщению других матричных алгоритмов на случай кольца коэффициентов [89].

В седьмой главе разработанная техника перенесена на случай большого простого поля [83, 84, 85, 86]. Такая задача является ядром алгоритма дискретного логарифмирования с факторной базой.

Разработан способ параллельных вычислений на системе с s узлами, для которой параллельная сложность (без пересылок) примет вид [83]

$$\mathcal{O}\left(\frac{dN^2}{s} + \frac{N^2}{q} + Ns\right), \tag{23}$$

где q - параметр алгоритма, а число пересылок линейно зависит от q. Этот результат получен в соавторстве с Н.Л.Замарашкиным. Конструкция использования приближений Паде и идея синхронизации, реализованные в виде формул, принадлежат М.А.Черепневу, а идея использования малой вероятности вырождения матриц над большим простым полем и упрощение алгоритма в этом случае принадлежат Н.Л.Замарашкину.

Получены два государственных свидетельства на регистрацию программного обеспечения на основе результатов исследований, изложенных в шестой и седьмой главах диссертации [87, 88].

Апробация работы.

Результаты работы неоднократно обсуждались на следующих

кафедрах МГУ им. М.В.Ломоносова:

— Теории чисел механико-математического факультета,

— Алгебры механико-математического факультета,

— Вычислительной математики механико-математического факультета,

— Математического анализа механико-математического факультета,

— Дискретной математики механико-математического факультета,

— Оптимального управления механико-математического факультета,

— Математической кибернетики факультета вычислительной математики и кибернетики,

в других организациях:

— в лаборатории по математическим проблемам криптографии МГУ им. М.В.Ломоносова,

— на кафедре компьютерного и математического моделирования Института математики, физики и информатики Тамбовского государственного университета им. Г.Р.Державина,

— в Академи криптографии РФ,

— в 8 Центре ФСБ РФ,

— в 16 Центре ФСБ РФ,

— в ОАО ”Концерн ”Автоматика”,

— в НПО "Квант"

— в Институте вычислительной математики РАН,

— в Математическом институте им. В.А.Стеклова РАН,

— в Институте проблем информатики РАН

на конференциях:

— Математика и безопасность информационных технологий (2003, 2004, 2007, 2008, МГУ, Москва),

— Проблемы теоретической кибернетики (2002, Казань; 1999, Нижний Новгород)

— Ломоносовские чтения (2005, 2008, МГУ, Москва).

— 12-th workshop on computer algebra (2008, Дубна),

— Applications of Computer Algebra (2008, Research Institute of Symbolic Computation, Austria),

— Колмогоровские чтения (2009, Тамбов),

— Parallel Computer Algebra (2010, Тамбов).

— Интеллектуальные системы компьютерные науки (2011, Москва)

Получен патент РФ на изобретение [43].

Часть III

Основное содержание диссертации

Глава 1

Показательные сравнения

Операция возведения в степень с вычислительной точки зрения достаточно проста. В то же время нахождение показателя по основанию и результату возведения в степень, то есть решение показательного сравнения, в случае если его рассматривать в конечном поле или группе точек на эллиптической кривой (дискретное логарифмирование), является достаточно сложной задачей. Это обстоятельство широко используют при построении асимметричных криптографических протоколов различного назначения. Наиболее часто используемый среди них — это протокол Диффи-Хеллмэна открытого распределения ключей. При помощи решения показательных сравнений этот протокол легко вскрывается. Однако является ли задача его вскрытия столь же трудной, как и задача решения соответствующего показательного сравнения, пока не ясно. Предпринятое в данной главе исследование свойств показательных сравнений проливает свет и на эту проблему.

1.1 О подъёме решений показательных сравнений

Сведение решения сравнения по модулю степени простого числа к случаю, когда эта степень равна единице, называется подъёмом решений.

В данном разделе приведены некоторые результаты о подъёме решений

показательных сравнений в кольцах вычетов [19, 20].

Логарифмическими будем называть функции натурального аргумента, удовлетворяющие уравнениям вида $f(x_1x_2) = f(x_1) + f(x_2)$, откуда при $m \in \mathbb{N}$ имеем $f(x^m) = mf(x)$.

Поскольку $f({x_1}^{\alpha_1} \cdot \dots \cdot {x_k}^{\alpha_k}) = \alpha_1 f(x_1) + \dots + \alpha_k f(x_k)$, то для того, чтобы задать логарифмическую функцию на некоторой группе, достаточно задать её на некоторой системе мультипликативных образующих.

Легко видеть, что такие функции, применённые к показательным уравнениям, дают линейные уравнения на показатель. В качестве примера можно привести функцию дискретного логарифмирования, называемую в более ранней литературе индексом. Её значение есть решение сравнения

$$a \equiv b^x \pmod m \tag{1.1}$$

$$x = \operatorname{ind}_b^m a = \operatorname{ind} a\,.$$

Функция индекс, очевидно, удовлетворяет сравнению

$$\operatorname{ind} a_1a_2 \equiv \operatorname{ind} a_1 + \operatorname{ind} a_2 \pmod{\varphi(m)}\,.$$

Рассмотрим функцию натурального аргумента, определяемую следующим образом:

$$L(p^\alpha) = \varphi(p^\alpha) = p^{\alpha-1}(p-1) \text{ при } p \geq 3\,,$$
$$L(2^\alpha) = 2^{\alpha-2} \text{ при } \alpha \geq 3\,,$$
$$L(4) = 2, L(2) = 1.$$
$$L(m) = \text{НОК}[L(p_i^{\alpha_i})], m = \prod p_i^{\alpha_i}.$$

Эта функция называется *функцией Кармайкла*. Можно показать, что для любого $a \in (\mathbb{Z}/m\mathbb{Z})^*$ справедливо сравнение $a^{L(m)} \equiv 1 \pmod m$, то есть $L(m)$ делится на порядок любого элемента этой группы. Кроме того, в группе $(\mathbb{Z}/m\mathbb{Z})^*$ существует элемент порядка $L(m)$. *Частным Ферма* называется функция

$$Q_m \pmod m : \quad (\mathbb{Z}/m^2\mathbb{Z})^* \to \mathbb{Z}/m\mathbb{Z},$$

где Q_m определена равенством

$$Q_m(a) = \tfrac{a^{L(m)}-1}{m}\,.$$

Число $T \mid m^2$ является *периодом* частного Ферма, если $Q_m(a+T) = Q_m(a) \pmod m$ для любого $a \in \mathbb{Z}$. В этом случае при $m \mid T$ можно считать, что $Q_m \pmod m$ определена на $(\mathbb{Z}/T\mathbb{Z})^*$, то есть

$$Q_m \pmod m : \quad (\mathbb{Z}/T\mathbb{Z})^* \to \mathbb{Z}/m\mathbb{Z}.$$

Хорошо известны следующие утверждения:

Предложение 1.1. *[18] Период функции Q_m (возможно, не наименьший), равен*

$$T(m) = \frac{m^2}{(m, L(m))}.$$

Предложение 1.2. *[18] Справедливо сравнение*

$$Q_m(ab) \equiv Q_m(a) + Q_m(b) \pmod m.$$

Отсюда, в частности, следует, что $Q_m(b^x) \equiv xQ_m(b) \pmod m$. Обозначим $x = [\log a]_\alpha \pmod{\varphi(p^\alpha)}$ – решение сравнения $a \equiv g^x \pmod{p^\alpha}$ предполагая, что оно существует.

Предложение 1.3. *[18] Пусть $p \geq 3$, g – первообразный корень $\pmod p^\alpha$, $\alpha \geq 2$. Тогда $[\log a]_\alpha$, в случае его существования, есть единственное по $\pmod{\varphi(p^\alpha)}$ решение системы из следующих двух сравнений:*

$$\begin{cases} x \equiv [\log a]_1 \pmod{p-1} \\ Q_{p^{\alpha-1}}(g)x \equiv Q_{p^{\alpha-1}}(a) \pmod{p^{\alpha-1}} \end{cases} \tag{1.2}$$

В случае когда g не является первообразным корнем $\pmod m$, применение частного Ферма затруднено тем, что возможно $Q(g) \equiv 0 \pmod m$. Однако эту проблему можно легко обойти.

Пусть $p \mid Q_p(g)$. Обозначим $l = \gamma_p(Q_p(g)) \in \mathbb{N}$, то есть $p^l \parallel Q_p(g)$ (то есть $p^l \mid Q_p(g), p^{l+1} \nmid Q_p(g)$), $u = ord_p g$. Тогда $\frac{p-1}{u} \in \mathbb{N}$ и для некоторых $S, T, r \in \mathbb{N}, p \nmid S, p \nmid T$ имеем

$$\begin{cases} g^{p-1} = 1 + p^{l+1}S \\ g^u = 1 + p^r T \end{cases}$$

Из второго равенства имеем $g^{p-1} = 1 + p^r \frac{p-1}{u} T + p^{2r} W$, где $W \in \mathbb{N}$. Теперь из первого равенства получим

$$p^{l+1}S = p^r \frac{p-1}{u} T + p^{2r} W,$$

откуда, ввиду простоты p, имеем $r = l + 1$. Поэтому $ord_{p^{l+1}} g = ord_p g = u$. Поскольку при $k \in \mathbb{N}$ выполнено $ord_{p^k} g \mid ord_{p^{k+1}} g$, то $ord_{p^k} g = u$ при любом $k \in \{1, 2, \ldots, l+1\}$. Последовательно возводя равенство

$$g^u = 1 + p^{l+1}T, p \nmid T$$

в степень p, по индукции, получим

$$g^{p^k u} = 1 + p^{l+1+k} T_k, k, T_k \in \mathbb{N}, p \nmid T_k.$$

Из этого равенства, поскольку $p \nmid T_k$, имеем $ord_{p^{l+1+k}} g = p^k v$, где $v \mid u$, поэтому $g^{p^k v} \equiv 1 \pmod p$ Применяя малую теорему Ферма, выводим, что $u = v$.

Таким образом, верна следующая

Лемма 1.1. *[19, 20] Пусть p – простое число, $(g, p) = 1, l = \gamma_p(Q_p(g)) \in \mathbb{N}$. Тогда*

$$ord_{p^\alpha} g = \begin{cases} ord_p g \text{ при } \alpha \in \{1, 2, \ldots, l\}, \\ p^{\alpha-(l+1)} ord_p g \text{ при } \alpha \geq l+1. \end{cases}$$

Теорема 1.1. *[19, 20] Пусть p – простое число, $(g,p)=1, l=\gamma_p(Q_p(g))\in \mathbb{N}, \alpha\in\mathbb{N}\setminus\{1\}$. Тогда, если сравнение*

$$a\equiv g^x \pmod{p^\alpha} \tag{1.3}$$

разрешимо, то его решение x удовлетворяет следующим условиям:

1. *При $\alpha\in\{1,\dots,l\}$ выполнено $ord_{p^\alpha}g = ord_p g$ и x есть единственное* $(\mathrm{mod}\ ord_p g)$ *решение сравнения*

$$a\equiv g^x \pmod{p} \tag{1.4}$$

2. *При $\alpha\geq l+1, k=\max\{l+1,\alpha-(l+1)\}$ выполнено равенство $ord_{p^\alpha}g = p^{\alpha-(l+1)}ord_p g$ и x есть единственное* $(\mathrm{mod}\ p^{\alpha-(l+1)}ord_p g)$ *решение системы*

$$\begin{cases} a\equiv g^x \pmod{p} \\ x\frac{Q_{p^{k-l}}(g)}{p^l}\equiv \frac{Q_{p^{k-l}}(a)}{p^l} \pmod{p^{\alpha-(l+1)}}. \end{cases} \tag{1.5}$$

Доказательство.

Равенства для $ord_{p^\alpha}g$ доказаны в лемме. Рассмотрим два случая.

1. Если x удовлетворяет сравнению (1.3), то, очевидно, оно также удовлетворяет сравнению (1.4), имеющему по модулю $ord_p g = ord_{p^\alpha}g$ единственное решение.

2. Если x удовлетворяет сравнению (1.3), то по модулю $ord_p g$ оно, как и в случае 1, однозначно определено равенством (1.4). В силу леммы, для доказательства теперь достаточно доказать второе сравнение из системы (1.5).

При $k\geq l+1$ введём новую функцию $Q_{p^k,l}(a)=\frac{Q_{p^{k-l}}(a)}{p^l}=\frac{a^{p^{k-(l+1)}(p-1)}-1}{p^k}$, определённую для a, таких, что $p^l \mid Q_{p^k,l}(a)$ или $ord_{p^k}a \mid p^{k-(l+1)}(p-1)$. Тогда, по предыдущей лемме, при всех $k\geq l+1$ число $Q_{p^k,l}(g)$ определено и не делится на p (иначе $ord_{p^{k+1}}g \mid p^{k-(l+1)}(p-1)$). Кроме того, для этой функции, так же как

и для обычного частного Ферма, выполнено логарифмическое свойство, то есть если $Q_{p^k,l}(a), Q_{p^k,l}(b)$ определены, то

$$Q_{p^k,l}(ab) \equiv Q_{p^k,l}(a) + Q_{p^k,l}(b) \pmod{p^k} \tag{1.6}$$

Из определения, применяя бином Ньютона, получим, что $Q_{p^k,l}(a+p^\alpha) \equiv Q_{p^k,l}(a) \pmod{p^{\alpha-(l+1)}}$, как только

$$2\alpha \geq k + \alpha - (l+1).$$

Но тогда $k \leq \alpha + l + 1$. Поэтому, применяя функцию $Q_{p^k,l}$ для $k = \max\{l+1, \alpha-(l+1)\}$ (см. 1.6) к сравнению (1.3), из сравнения (1.6) получим, что

$$xQ_{p^k,l}(g) \equiv Q_{p^k,l}(a) \pmod{p^{\alpha-(l+1)}},$$

где $(Q_{p^k,l}(g), p) = 1$.

Теорема доказана.

Приведённые рассуждения не опираются на существование первообразного корня, а только на существование решения соответствующего сравнения.

1.2 Сравнительная стойкость протокола Диффи-Хеллмэна и задачи дискретного логарифмирования

Для вскрытия схемы Диффи-Хеллмэна криптоаналитик должен решить следующую задачу: по известным p, g, g^a, g^b найти g^{ab}. Эта задача носит название задачи Диффи-Хеллмэна. Она может быть решена при помощи дискретного логарифмирования. Может ли она быть решена без использования дискретного логарифмирования, неизвестно. Для практики важно выявить значения параметров, при которых задача Диффи-Хеллмэна столь же трудна, как и задача дискретного логарифмирования.

Число M называется B-гладким, если для любого простого q, такого, что $q|M$, выполнено $q \leq B$.

Будем называть рассматриваемую задачу дискретного логарифмирования сертифицированной, если разложение $p-1, p_i - 1, \quad i = 1, \ldots, r, q_{ij} - 1$ для простых $q_{ij} \mid p_i$ и так далее на простые множители, а также первообразные корни по модулям $p, p_i, \quad i = 1, \ldots, r, q_{ij}$ — известны.

В следующей теореме приведён пример значений модуля p, по которому проводятся вычисления, для которых задачи дискретного логарифмирования и Диффи-Хеллмэна полиномиально эквивалентны.

Теорема 1.2. *[91] Пусть $\varphi(p-1)$ – B-гладкое число, где B - наперед заданная константа. Тогда сертифицированная задача дискретного логарифмирования может быть решена за время, равное полиному от длины записи числа p, умноженному на время решения задачи Диффи-Хеллмэна.*

Теорема останется верной, если $B = P(|p|)$, где $P(\cdot)$ – некоторый полином, $|p|$ - длина записи числа p.

Связь между сложностями задач дискретного логарифмирования и Диффи-Хеллмэна может быть установлена в общем случае. Введём следующие обозначения. Пусть $G(t, m)$ — произвольная коммутативная циклическая группа порядка m с операцией, которую будем в дальнейшем обозначать $+$, требующей для своего выполнения t битовых операций, и пусть $L(t, m)$ обозначает минимальное количество битовых операций, необходимых для решения задачи дискретного логарифмирования в группе $G(t, m)$, то есть при известных $a, b \in G(t, m)$ найти $n \in Z_m$, такое, что

$$a = nb, \tag{1.7}$$

здесь nb есть $b + \cdots + b$, где элемент b повторяется n раз.

Не ограничивая общности дальнейших рассуждений, будем считать b образующим

группы $G(t,m)$.

Пусть $D(t,m)$ — неубывающая по m оценка количества битовых операций, необходимых для решения задачи Диффи-Хеллмэна: при известных a_1, a_2 и b таких, что $a_1 = n_1 b, a_2 = n_2 b$, найти

$$a_3 = (n_1 n_2) b. \tag{1.8}$$

Пусть $D^*(t,m)$ — также неубывающая по m оценка количества битовых операций, необходимых для решения той же задачи при помощи алгоритмов, количество битовых операций в которых удовлетворяет неравенству

$$D^*(t,m) \leq tD^*(C,m),$$

для некоторой абсолютной константы C (например таких, которые используют только операции, сложность которых не более, чем сложность групповой операции).

Прежде чем сформулировать теорему, сделаем несколько замечаний. Пусть

$$p - 1 = \prod_{i=1}^{r} {p_i}^{\alpha_i}.$$

Замечание. *Маккарли в [92] доказал, что, в случае* $r = 3, p_1 = 2$ *и* $\alpha_2 = \alpha_3 = 1$, *количество битовых операций, необходимых для разложения на простые множители числа* $p-1$, *есть* $O(D(t,p-1) + \log^2(p-1))$ *и не превосходит*

$$D(t,p-1) \cdot \log^2(p-1),$$

если константу из O, также как и все абсолютные константы в дальнейшем, считать учтённой в основании логарифма. Принимая во внимание тот факт, что для всех известных алгоритмов разложения на множители этот случай представляется наиболее трудоёмким, эта оценка кажется правдоподобной и для $p-1$, разлагающихся на большее число простых множителей.

Замечание. *Поиск первообразного корня по модулю p, как было описано в первой главе, можно осуществить следующим образом. Количество первообразных корней по модулю p равно $\varphi(p-1)$, где φ - функция Эйлера. Поэтому, принимая во внимание известную оценку Ландау [93]*

$$\varphi(n) \geq C\frac{n}{\log\log n},$$

где C — некоторая абсолютная константа, можно считать эффективной случайную выборку $g \in \{1, \ldots, p-1\}$ с последующей проверкой выполнения неравенств

$$g^{\frac{p-1}{p_i}} \not\equiv 1 \pmod{p}, \qquad i = 1, \ldots, r. \tag{1.9}$$

В случае успеха, число g является первообразным корнем по модулю p. Количество битовых операций, необходимых для проверки соотношений (1.9), не превосходит $\log p \cdot u(p) \cdot \nu(p-1)$, где $\nu(p-1)$ - количество различных простых делителей числа $p-1$, а $u(p)$ - число битовых операций, необходимых для умножения по модулю p. Несложные вычисления показывают, что, действуя таким способом, за $\log p \log\log p \cdot u(p) \cdot \nu(p-1)$ битовых операций, мы найдём первообразный корень по модулю p с вероятностью $1 - e^{-C} + o(1)$ при $p \to \infty$. Повторяя эту операцию конечное, но большое число раз, получим вероятность, близкую к единице. Кроме того, применяя преобразование Абеля, находим, что

$$\log(2 \cdot 3 \cdot 5 \cdots p_s) = s\log s + o(s\log s),$$

где p_s - s-е простое число, поэтому

$$\nu(p-1) = \nu(P) = O\left(\frac{\log P}{\log\log P}\right) \leq O\left(\frac{\log p}{\log\log p}\right),$$

где

$$P = \prod_{i=1}^{\nu(p-1)} p_i$$

Таким образом, первообразный корень $\pmod p$ можно найти за $u(p)\cdot\log^2 p$ арифметических операций с вероятностью, близкой к единице.

Оба эти замечания касаются случая несертифицированной задачи дискретного логарифмирования. В случае же сертифицированной задачи разложение $p-1, p_i-1, \quad i=1,\dots,r, q_{ij}-1$ для простых $q_{ij}\mid p_i$ и так далее на простые множители, а также первообразные корни по модулям $p, p_i, \quad i=1,\dots,r, q_{ij}$ и т.д. считаются известными.

Для произвольного $m=\prod_{i=1}^r {p_i}^{\alpha_i}$ рассмотрим разложения на простые множители чисел $p_i-1, \quad i=1,\dots,r$, разложения $q_{ij}-1$ для простых $q_{ij}\mid p_i$, и так далее. Получившееся разветвление называется деревом Пратта числа m, а встречающиеся простые числа - его узлами. Свяжем узлы q_{ij} с узлом p_i. Назовём ветвью максимальную последовательность связанных друг с другом узлов. Обозначим $s=s(m)$ длину наибольшей ветви дерева m.

Следующая теорема устанавливает связь между сложностями задач дискретного логарифмирования и Диффи-Хеллмэна в общем случае.

Теорема 1.3. *[23] Для сертифицированной задачи дискретного логарифмирования*

$$L(t,m)\le s\log^2 m D(\dots D(D(t,m),m)\dots), \tag{1.10}$$

где справа стоит s - кратная итерация функции $D(t,m)$,

$$L(t,m)\le ts\log^2 m(D^*(C,m))^s. \tag{1.11}$$

Теорема 1.4. *[23] В случае, если оценка из приведённого выше замечания для числа операций, необходимых для разложения на множители, выполняется для любого простого p, то утверждение теоремы 1.3 останется верным и без условия сертифицированности, но только для вероятностных алгоритмов дискретного логарифмирования.*

Подобная оценка с $s = 1$, но только для простых m, удовлетворяющих некоторому условию гладкости, была получена ранее в теореме 1.2 (см. также [94]).

Доказательство теоремы 1.3.

Для простоты рассмотрим сначала случай, когда m - простое число, $m = p$. Без ограничения общности можно считать, что в уравнении (1.7) $n \not\equiv 0 \pmod p$, а b - не единичный элемент группы $G(t,p)$. Введём на множестве элементов $\{nb \mid n \not\equiv 0 \pmod p\}$ новую групповую операцию, заданную равенством

$$n_1 b \oplus n_2 b = (n_1 n_2) b.$$

Количество битовых операций, необходимых для выполнения этой операции, равно, очевидно, $D(t,p)$. Так как отличные от нуля вычеты по простому модулю образуют циклическую группу по умножению, мы получаем циклическую группу порядка

$$\varphi(p) = p - 1 = \prod_{i=1}^{r} {p_i}^{\alpha_i}$$

с аддитивной операцией $\oplus$, где $p_i, \quad i = 1, \ldots, r$, - различные простые числа, а $r = \nu(p-1)$. Обозначим эту группу $G(D(t,p), p-1)$. Ясно, что b - единичный элемент этой группы.

Пусть g — первообразный корень по модулю p, тогда gb - образующий группы $G(D(t,p), p-1)$. Эта группа является прямой суммой примарных циклических групп порядка ${p_i}^{\alpha_i}$ с образующими соответственно $g_i b$, где

$$g_i \equiv g^{\frac{p-1}{{p_i}^{\alpha_i}}} \pmod p, \qquad i = 1, \ldots, r.$$

Пусть $n = g^v \pmod p$. Уравнение (1.7) можно переписать в виде

$$a = \bigoplus_i g_i^{n_i} b = {g_1}^{n_1} b \oplus \cdots \oplus {g_r}^{n_r} b,$$

где

$$1 \le n_i \le {p_i}^{\alpha_i}, \quad v \equiv \frac{p-1}{{p_i}^{\alpha_i}} n_i \pmod{{p_i}^{\alpha_i}}, \quad i = 1, \ldots, r,$$

или

$$a = n_1 \cdot (g_1 b) \oplus \cdots \oplus n_r \cdot (g_r b),$$

где операция $\cdot$ (операция умножения на натуральные числа элементов группы $G(D(t,p), p-1)$ определена для операции $\oplus$ так же, как для $+$, а именно

$$l \cdot (kb) = kb \oplus \cdots \oplus kb = k^l b,$$

где операция $\oplus$ связывает l элементов. Заметим, что l при этом можно считать вычетом по модулю $p-1$. Количество битовых операций, необходимых для такого умножения с применением бинарного алгоритма, очевидно, не превосходит $\log n \cdot D(t,p)$. Теперь, чтобы определить число n в выражении (1.7), будем искать числа n_i, а затем найдём n по формуле

$$n \equiv \prod_{i=1}^{r} {g_i}^{n_i} \pmod{p}.$$

Для каждого $i = 1, \ldots, r$ с помощью бинарного алгоритма найдём

$$a_i = \frac{p-1}{{p_i}^{\alpha_i}} \cdot a = \frac{p-1}{{p_i}^{\alpha_i}} n_i \cdot (g_i b).$$

Для этого потребуется не более, чем $\nu(p-1)\log(p-1)D(t,p)$ битовых операций. Ещё $\nu(p-1)\log(p-1)(u(p)+t)$ битовых операций необходимо для вычисления $g_i, g_i b$ при известных g и b для всех $i = 1, \ldots, r$. Затем, решая задачу дискретного логарифмирования в циклической подгруппе порядка ${p_i}^{\alpha_i}$ группы $G(D(t,p), p-1)$, порождённой элементом $g_i b$, найдём такое t_i, что

$$a_i = t_i \cdot (g_i b). \tag{1.12}$$

Для этого потребуется $L(D(t,p), {p_i}^{\alpha_i})$ битовых операций. Причём, очевидно, что

$$n_i \frac{p-1}{{p_i}^{\alpha_i}} \equiv t_i \pmod{{p_i}^{\alpha_i}}.$$

С помощью алгоритма Евклида элемент $\frac{p-1}{{p_i}^{\alpha_i}}$ по модулю ${p_i}^{\alpha_i}$ можно обратить за $u(p)\log {p_i}^{\alpha_i}$ битовых операций, поэтому количество битовых операций, необходимых для нахождения каждого n_i при известных $a_i, g_i b$, не превосходит

$$L(D(t,p),{p_i}^{\alpha_i}) + u(p)\log {p_i}^{\alpha_i}. \tag{1.13}$$

Уравнение (1.12) можно решать и по-другому, а именно, - домножая его на ${p_i}^{\alpha_i - 1}$, не более, чем за $L(D(t,p),p_i)$ битовых операций, получим t_i по модулю p_i.

Пусть $t_i = t_{i0} + p_i t_{i1}, \qquad 0 \le t_{i0} < p_i,$ тогда $a_i = (t_{i0} + p_i t_{i1}) \cdot (g_i b),$ а так как при перемножении показатели складываются, то

$$a_i \oplus (p_i - t_{i0}) \cdot (g_i b) = (t_{i1} + 1) \cdot ({g_i}^{p_i} b).$$

Число t_{i1} по модулю ${p_i}^{\alpha_i - 1}$ определяется не более, чем за $L(D(t,p),{p_i}^{\alpha_i - 1})$ битовых операций. Таким образом, в равенстве (1.13) можно заменить $L(D(t,p),{p_i}^{\alpha_i})$ на

$$\alpha_i(\log p_i D(t,p) + L(D(t,p),p_i)).$$

Суммируя сказанное, получим следующую рекуррентную формулу для оценки сложности сертифицированной задачи дискретного логарифмирования:

$$L(t,p) \le D(t,p)\log^2 p + \sum_{{p_i}^{\alpha_i} \| p-1} \alpha_i L(D(t,p),p_i) \tag{1.14}$$

(считаем, что $D(t,p) \ge \max\{t, u(p)\}$).

Повторяя те же рассуждения с заменой p на p_i и т.д., мы, в конечном счёте, придём к задаче дискретного логарифмирования в группе порядка 2, которая может быть решена перебором.

Утверждение теоремы докажем индукцией по s. Основание индукции следует из неравенства (1.14) и тривиальной оценки

$$L(t,p) \le tp. \tag{1.15}$$

При рассмотрении шага индукции воспользуемся неравенством (1.14) и неравенством (1.10) с заменой s на $s-1$. Тогда получим, что

$$L(t,p) \le D(t,p)\log^2 p + \sum_{{p_i}^{\alpha_i}\|p-1} \alpha_i(s-1)\log^2 p_i D(\dots D(D(t,p),p)\dots)$$

$$\le D(\dots D(D(t,p),p)\dots)\left(\log^2 p + (s-1)\log p \sum_{{p_i}^{\alpha_i}\|p-1} \alpha_i \log p_i\right)$$

$$\le s\log^2 p D(\dots D(D(t,p),p)\dots),$$

где в формулах стоят s-кратные итерации функции $D(t,p)$, причём по определению $D(x,y)$ не убывает по y.

Случай произвольного

$$m = \prod_{i=1}^{r} {q_i}^{\alpha_i}$$

сводится к рассмотренному выше умножением равенства (1.7) на $\frac{m}{q_i}$ и т.д., аналогично приведённому выше решению уравнения (1.12).

Доказательство неравенства (1.11) очевидно. Теорема 1.3 доказана.

Доказательство теоремы 1.4 очевидным образом следует из оценки (1.14) и приведённого выше замечания.

Отметим, что в качестве $D(x,y)$ в неравенство (1.10) (или $D^*(x,y)$ в неравенство (1.11)) может быть подставлена оценка числа битовых операций любого массового алгоритма (то есть такого, который может быть применён к любой циклической группе порядка y с групповой операцией, требующей x битовых операций для своего выполнения).

Для числа s тривиальная оценка $s(m) \le \log m$ для подстановки в полученные неравенства не подходит, так как при этом, пренебрегая, как и выше, изменением оснований, используемых логарифмов при ограниченном t, получим оценку:

$$L(t,m) \le (D^*(C,m))^{\log m}\log^3 m = e^{\log(D^*(C,m))\log m}\log^3 m = m^{\log(D^*(C,m))}\log^3 m.$$

Эта оценка, очевидно, хуже тривиальной оценки (1.15), где p заменено на m.

Из теоремы Линника (см. [93] с.411) следует, что для некоторой абсолютной константы C и любого простого числа p наименьшее простое число вида $1+pn$ не превосходит p^C. Поэтому существует бесконечная последовательность натуральных чисел m, для которых

$$s(m) \geq \log\log m.$$

При $s(m) \leq \alpha(m)\frac{\log m}{\log\log m}, D^*(C,m) \leq \log m$ и ограниченном t из оценки (1.11) получим, что

$$L(t,m) \leq m^{\alpha(m)}\log^3 m,$$

где

$$\frac{\log\log m}{\log m} \ll \alpha(m) \ll \log\log m,$$

что при

$$\alpha(m) = o(1)$$

даёт нетривиальную информацию о связи между сложностями задач дискретного логарифмирования и вскрытия схемы Диффи-Хеллмэна.

Некоторые результаты, близкие к задаче оценки величины $s(m)$, приведены в следующем подразделе (см. также [29]). Задача выявления классов простых чисел, для которых $\alpha(p) = o(1)$, остаётся открытой. Отметим также, что простые Ферма F_n среди остальных простых чисел определяются равенством $s(F_n) = 1$.

В 2008 году, K. Ford, С.В.Конягиным, F.Luca [28] для почти всех натуральных чисел m выполнено $clog_2log_2m \leq s(m) \leq (logm)^{1-0.0378}$. В приведённых выше обозначениях это означает, что $\alpha(m) \leq \frac{loglogm}{(logm)^{0,0378}}$, откуда $L(m) \leq exp\{(logm)^{1-0,0378}(loglogm + c)\}$. Это пока ещё хуже субэкспоненциальных оценок известных алгоритмов дискретного логарифмирования. В этой же работе авторы выдвинули гипотезу о том, что для почти всех натуральных чисел $s(m) \leq clog_2log_2m$ для некоторой константы c. Из этой гипотезы и результатов этого раздела следует, что, в случае существования общего полиномиального алгоритма решения задачи Диффи-Хеллмэна, существует алгоритм решения

задачи дискретного логарифмирования со сложностью $exp\{C(log_2 log_2 m)^2\}$, что уже очень близко к полиномиальной оценке, и, конечно, намного быстрее существующих алгоритмов дискретного логарифмирования [95].

1.3 О полиномиальной эквивалентности задач дискретного логарифмирования и Диффи-Хеллмэна на некоторых стандартных кривых

Описанный выше алгоритм решения задачи дискретного логарифмирования с оракулом Диффи-Хеллмэна может быть применён к атаке на государственный стандарт ГОСТ Р34.10-2012 эллиптических кривых, используемых в современных системах защиты информации.

Применим описанные выше рассуждения к группе точек на эллиптической кривой $E[\mathbb{F}_r]$ над конечным полем $\mathbb{F}_r$ из простого r числа элементов.

Пусть используемая циклическая группа точек $< P >$ кривой имеет простой порядок p. Пусть $p-1 = \prod_{i=1}^{t} {q_i}^{\alpha_i}$. В стандарте ГОСТ Р34.10-2012 не указано никаких требований на это разложение, поэтому мы будем считать его имеющим общий вид. А именно, пусть

$$\psi(x,y) = \sharp\{1 \le n \le x \mid \text{для любого } q, \text{ простого делителя } n, \text{ имеем: } q < y\},$$

тогда при фиксированном $u \ge 3, x \ge 1$ ([105] Теорема 3.1)

$$\psi(x, \sqrt[u]{x}) \ge \frac{x}{u^{u(1+o(1))}}. \tag{1.16}$$

Таким образом с вероятностью, равной константе можно считать, что $q_i \le \sqrt[u]{p}$ для фиксированного u.

Для группы точек на эллиптической кривой $E[\mathbb{F}_r]$ над конечным полем $\mathbb{F}_r$ из простого r числа элементов, обозначим $[n]P$ сумму n точек P.

Пусть по двум точкам $Q, P \in E[\mathbb{F}_r]$, связанным равенством

$$Q = [n]P,$$

надо найти n, определённое по модулю $p = ordP$. Обозначим сложность этой задачи $DLE(r,p)$, а сложность вычисления $[n_1 n_2]P$ по паре $([n_1]P, [n_2]P)$ обозначим $DHE(r,p)$. Будем полагать, что эта последняя задача не проще одной операции на эллиптической кривой $E[\mathbb{F}_r]$. Символом log будем обозначать логарифм по некоторому фиксированному основанию, значение которого каждый раз будет некоторой эффективной абсолютной константой.

Теорема 1.5. *[82]* $DLE(r,p) \leq DHE(r,p) logp \sum_{i=1}^{t} {q_i}^{\alpha_i}$.

Доказательство. Очевидно, что $n \equiv 0(modp)$ тогда и только тогда, когда Q является бесконечно удалённой точкой. В противном случае $n \equiv g^h(modp)$, где g - первообразный корень по модулю p, а h определено по $(mod(p-1))$. Далее вместо n будем искать h.

Отметим, что множество точек $[n]P, n \not\equiv 0(modp)$ образует циклическую группу порядка $p-1$ относительно операции

$$[n_1]P * [n_2]P = [n_1 n_2]P,$$

которую будем обозначать G.

Пусть $q \mid p-1$. Будем перебирать случайные $j_0', j_0'' \in \{1, \ldots, \frac{p-1}{q}\}$ и проверять, существует ли такое $i \in \{0, 1, \ldots, q-1\}$, что

$$[g^{j_0'}]Q = [g^{j_0''}][g^{\frac{p-1}{q}i}]P, \tag{1.17}$$

Если да, то

$$n \equiv g^{j_0'' + p - 1 - j_0' + \frac{p-1}{q}i}(modp),$$

или

$$h \equiv j_0'' - j_0' + \frac{p-1}{q}i(mod(p-1)),$$

или

$$h \equiv j_0'' - j_0'(\mathrm{mod}(\frac{p-1}{q}).$$

Для построения

$$[g^{j_0'}]Q, [g^{j_0''}]P \tag{1.18}$$

достаточно $4log_2 p \leq logp$ операций сложения на эллиптической кривой.

По известной теореме о парадоксе дней рождения [106] среднее число пар $(j_0', j_0''), j_0', j_0'' \in \{1, \ldots, \frac{p-1}{q}\}$, которые необходимо проверить прежде, чем будет получен положительный ответ на вопрос о существовании i, равно $(\sqrt{\pi\frac{p-1}{q}} + O(1))^2$. При этом j_0' и j_0'' пробегают некоторые случайные независимые подмножества в $\{1, \ldots, \frac{p-1}{q}\}$ объёма $\sqrt{\pi\frac{p-1}{q}} + O(1)$. Отметим, что в этом месте можно заменить случайный перебор на $j_0'' = 0, j_0' \in \{1, \ldots, \frac{p-1}{q}\}$, что увеличит объём рассматриваемых подмножеств но не изменит число проверок.

Если теперь вместо q последовательно подставить $\tilde{q}_j$ такие, что $\mathrm{НОК}[\frac{p-1}{\tilde{q}_j}] = p-1$ и воспользоваться Китайской теоремой об остатках, то получим $h(mod(p-1))$. То есть среднее число проверок есть

$$O(\sum_j \pi\frac{p-1}{\tilde{q}_j}),$$

что при $\tilde{q}_j = \frac{p-1}{q_j{}^{\alpha_j}}, j = 1, \ldots t$, даёт

$$O(\sum_{i=1}^{t} q_i{}^{\alpha_i}). \tag{1.19}$$

Равенство (1.17) означает, что

$$Q_{j_0'+p-1-j_0''} = [g^{j_0'+p-1-j_0''}]Q \in G_{0,\frac{p-1}{q}} = \{[g^{\frac{p-1}{q}i}]P \mid i = 0, 1, \ldots, q-1\}. \tag{1.20}$$

Вычисление $Q_{j_0'+p-1-j_0''}$ при известных j_0', j_0'' осуществляется квадратичным алгоритмом и требует не более $2log_2 p$ умножений по $(modp)$ и $2log_2 p$ сложений

точек на эллиптической кривой, что предполагается сложнее (соответствующие формулы требуют 5 умножений в поле на одно сложение на эллиптической кривой).

Отметим, что $G_{0,\frac{p-1}{q}}$ - циклическая подгруппа группы G порядка q. Введём операцию "$\cdot$"(см. [23])

$$k \cdot [n]P = \underbrace{[n]P * \cdots * [n]P}_{k} = [n^k]P.$$

Тогда вопрос о принадлежности (1.20) является вопросом о разрешимости задачи дискретного логарифмирования в циклической группе G. Поскольку группа порядка q в такой группе единственна, то эта принадлежность выполнена тогда и только тогда, когда

$$\underbrace{Q_{j_0'+p-1-j_0''} * \cdots * Q_{j_0'+p-1-j_0''}}_{q} = [1]P = P, \tag{1.21}$$

что проверяется со сложностью $2DHE(r,p)\log_2 q$. Теорема доказана.

В общем случае можно считать, что (см. [23])

$$DLE(r,p) \leq O(s(p)log^2 p(DHE(r,p))^{s(p)}),$$

где $s(p)$ - длина наибольшей ветви дерева Пратта [24], для которой на сегодняшний день имеется только тривиальная оценка $s(p) \leq log_2 p$ (см. некоторое развитие в статье [110]).

Доказанная теорема показывает в частности, что на подруппах точек простого порядка p на эллиптических кривых, удовлетворяющих стандарту ГОСТ Р34.10-2012, где $p-1$ раскладывается на "маленькие"простые множители, задачи Диффи-Хеллмэна и дискретного логарифмирования полиномиально эквивалентны.

Глава 2

Некоторые свойства распределения простых чисел

Для оценки стойкости многих криптографических протоколов требуется сравнить стойкости задач дискретного логарифмирования и Диффи-Хеллмэна [96]. В работе [23] (см. выше) эта задача сведена к оценке величины $s = s(m)$ - длины наибольшей ветви дерева Пратта числа m. В данном параграфе изучаются свойства разложения чисел вида $p-1$ на простые множители, которые, ввиду определения, связаны с величиной s.

Для оценки $s(m)$ для почти всех m можно использовать результаты о количестве различных простых делителей чисел вида $p-1$. Оценка снизу для этого количества даёт оценку сверху для самих простых делителей, которую в дальнейшем можно использовать для оценки $s(m)$.

В своих работах Харди и Рамануджан [98], а также Туран [99] показали, что за исключением $\overline{o}(x)$ натуральных значений $n, n \leq x$ выполняются неравенства

$$(1-\varepsilon)lnlnn < v(n) < (1+\varepsilon)lnlnn,$$

где $v(n)$ — количество различных простых делителей числа n. Аналогичный результат для чисел вида $p-1$, где p - простое число, получил Эрдёш [100]. В данном разделе похожий результат получен для количества простых делителей чисел $p-1$, больших некоторой растущей границы. Заметим, что, ввиду

очевидной оценки $s(n) \leq logn$, "маленькие" простые делители не оказывают влияния на величину $s(m)$.

Обозначим $v_{>t}(n)$ количество различных простых делителей числа n, больших t. Обозначим $N(M)$ количество элементов в множестве M.

Следующая теорема показывает, что для почти всех простых p половина простых делителей чисел $p-1$ достаточно велика.

Теорема 2.1. *[29] Для любого* $\varepsilon \in (0, \frac{1}{3})$, *и* $t = x^{\frac{1-\varepsilon}{2e}}$

$$N(p \leq x \mid v(p-1) \in [(1-\varepsilon)lnlnx, (1+\varepsilon)lnlnx], v_{>t}(p-1) > \frac{1}{2}v(p-1)) = \frac{x}{lnx} + \overline{o}(\frac{x}{lnx}).$$

Доказательство. Рассмотрим множество

$$M = (p \leq x | v(p-1) \in [(1-\varepsilon)lnlnx, (1+\varepsilon)lnlnx]).$$

По теореме 7.2 на странице 188 [93] $N(M) = \frac{x}{lnx} + \overline{o}(\frac{x}{lnx})$. Поэтому для доказательства нашей теоремы достаточно установить, что

$$N(p \in M | v_{\leq t}(p-1) > \frac{1}{2}v(p-1)) = \overline{o}(\frac{x}{lnx}).$$

Как доказано на странице 189 [93], число простых p, не превосходящих x, для которых все простые делители $p-1$ не превосходят $y = e^{\frac{lnx}{lnlnx}}$ есть $\overline{o}(\frac{x}{lnx})$. Значит, из нашего рассмотрения их можно отбросить. Поэтому рассматриваемую величину можно оценить как

$$\sum N(p \leq x | p, \frac{p-1}{n} \text{ - простые}) < \\ < \sum N(r < \frac{x}{n} | r, rn+1 \text{- простые}),$$

где суммирование ведётся по всем натуральным n, удовлетворяющим неравенствам $n < \frac{x}{y}, v_{\leq t}(p-1) > \frac{1}{2}v(p-1), v(p-1) \in [(1-\varepsilon)lnlnx, (1+\varepsilon)lnlnx]$. Слагаемые в последней сумме можно оценить по теореме 4.2 на странице 54 [93] при $s = 2$. Получим:

$$< \sum \frac{xn}{nln^2\frac{x}{n}\varphi(n)} <$$
$$< cx \sum \frac{1}{\varphi(n)ln^2\frac{x}{n}} <$$
$$< \frac{cx}{ln^2 y} \sum \frac{1}{\varphi(n)}.$$

Здесь суммирование ведётся по тем же натуральным n. Далее воспользуемся мультипликативностью функции Эйлера и известными оценками сумм по простым числам (см. [93] теорема 4.1. на странице 28).

$$\leq \frac{cx}{ln^2 y} \sum \frac{\left(\sum_{p<x} \sum_{i=1}^{\infty} \frac{1}{\varphi(p^i)}\right)^{k-\frac{1-\varepsilon}{2}lnlnx} \left(\sum_{p\leq t} \sum_{i=1}^{\infty} \frac{1}{\varphi(p^i)}\right)^{\frac{1-\varepsilon}{2}lnlnx}}{(k-\frac{1-\varepsilon}{2}lnlnx)!(\frac{1-\varepsilon}{2}lnlnx)!}$$

Здесь $a! = [a]!$, а суммирование ведётся по $k \in [(1-\varepsilon)lnlnx, (1+\varepsilon)lnlnx]$.

$$< \frac{cx}{(\frac{1-\varepsilon}{2}lnlnx)! \ln^2 y} \sum_{k\in[(\frac{1-\varepsilon}{2})lnlnx, (\frac{1}{2}+\frac{3}{2}\varepsilon)lnlnx]} \frac{1}{k!}(lnlnx + c_1)^k (lnlnt + c_2)^{\frac{1-\varepsilon}{2}lnlnx} \leq$$

для некоторых абсолютных постоянных c_1, c_2. Применяя формулу Стирлинга, получим для $\varepsilon_1 = \frac{1}{2} - \frac{3\varepsilon}{2} > 0$ следующую оценку:

$$\leq \frac{cx}{ln^2 y}(\frac{2e}{1-\varepsilon}\frac{lnlnt + c_2}{lnlnx})^{\frac{1-\varepsilon}{2}lnlnx} \sum_{k\leq(1-\varepsilon_1)lnlnx} \frac{(lnlnx + c_1)^k}{k!}$$

Теперь воспользуемся оценкой для получившейся суммы со страницы 191 [93]. Получим:

$$\leq \frac{cxlnln^2 x ln^{1-\delta} x}{ln^2 x}(\frac{2e}{1-\varepsilon}\frac{lnlnt + c_2}{lnlnx})^{\frac{1}{2}lnlnx}$$

Последнее при выбранном в условии t есть $\overline{o}(\frac{x}{lnx})$.

Следующая оценка снизу функции Эйлера улучшает известную оценку Ландау для почти всех натуральных значений аргумента.

Теорема 2.2. *[31] Для некоторой абсолютной константы* c

$$N(n \leq x | \varphi(n) > \frac{cn}{lnlnlnn}) = x + \overline{o}(x)$$

Доказательство. Как уже отмечалось в начале,

$$N(n \leq x | v(n) \in [(1-\varepsilon)lnlnx, (1+\varepsilon)lnlnx]) = x + \overline{o}(x).$$

Докажем, что для этих n выполняется нужная оценка.

$$\varphi(n) = n\prod_{p|n}(1-\frac{1}{p}) \geq n\prod_{i=1}^{v(n)}(1-\frac{1}{p_i}) = n\prod_{p \leq p_{v(n)}}(1-\frac{1}{p}) >$$

По теореме 4.1 на странице 28 [93], для некоторой абсолютной константы c_1 получим оценку:

$$> c_1\frac{n}{lnp_{v(n)}} > \frac{c_2 n}{\ln(v(n)lnv(n))} > \frac{cn}{lnlnlnn}.$$

Далее в этом разделе получены ассимптотики количества простых делителей чисел $q-1$, где q - большой простой делитель числа $p-1$.

Заметим, что, ввиду очевидной оценки $s(n) \leq log_2 n$, маленькие простые делители не оказывают влияния на величину $s(m)$.

Итак, $N(M)$ - количество элементов в множестве M.

Следующая теорема показывает, что для почти всех простых p числа $q-1$, где q — "большие" простые делители числа $p-1$, имеют достаточно много простых делителей.

Теорема 2.3. *[29] Для любого* $\varepsilon > 0$

$$N\left(p \le x \,\middle|\, \begin{array}{c} \textit{для всех простых } q > e^{\frac{\ln x}{\ln\ln x}} \textit{таких, что } \dfrac{p-1}{q} = n \in \mathbb{N}, \\ \textit{выполнено } v(q-1) \in \left[(1-\varepsilon)lnln\dfrac{x}{n}, (1+\varepsilon)lnln\dfrac{x}{n}\right] \end{array}\right) =$$

$$(1+\overline{o}(1))\frac{x}{lnx}.$$

Докажем сначала следующую лемму.

Лемма 2.1. *[29] Найдутся абсолютные положительные константы* C_1, C_2, *такие, что для любых* $k, n \in \mathbb{N}$

$$N\left(p \le x \,\middle|\, \frac{p-1}{n} = q \text{ - } \textit{простое, } v(q-1) = k\right)$$
$$< C_1 \frac{x(n+1)(lnln\frac{x}{n} + C_2)^{k+2}}{(k-1)!\varphi(n(n+1))ln^3\frac{x}{n}} + \overline{o}\left(\frac{x}{n(ln\frac{x}{n})^{\frac{1}{2}lnlnln\frac{x}{n}}}\right),$$

при $\frac{x}{n} \to \infty$, *причём* $\overline{o}$ *не зависит от* k *и от* n.

Доказательство. Обозначим $\frac{p-1}{n} = q \le \frac{x-1}{n} = f(x)$. Рассмотрим множество M_1 чисел $p \le x$, для которых $\frac{p-1}{n}$ - простое и любой простой делитель числа $\frac{p-1}{n} - 1$ меньше $y(f(x))$, где $y(z) = e^{\frac{lnz}{lnlnz}}$. Обозначим $N(x,y)$ число натуральных чисел, не превосходящих x, все простые делители которых не превосходят y. Для оценки этой величины применим известную лемму Ранкина [93, Лемма 5.2, §5, глава V]. Получим:

$$N(M_1) < N\left(q \text{ - простое} \;\middle|\; q \le f(x), \quad \begin{cases} r \mid q-1 \\ r\text{- прост} \end{cases} \Rightarrow r \le y(f(x))\right)$$

$$\le N\left(m \text{ - натуральное} \;\middle|\; m \le f(x), \quad \begin{cases} r \mid m \\ r\text{- прост} \end{cases} \Rightarrow r \le y(f(x))\right)$$

$$\leq N(f(x), y(f(x))) < f(x)e^{-\frac{ln_2(\frac{lnf(x)}{ln_2f(x)})ln_2f(x)}{lnf(x)}lnf(x)+ln(\frac{lnf(x)}{ln_2f(x)})(1+\overline{o}(1))}$$
$$= \overline{o}(f(x)e^{-\frac{1}{2}ln_3f(x)ln_2f(x)}) = \overline{o}(\frac{x}{n}e^{-\frac{1}{2}ln_3(\frac{x}{n})ln_2(\frac{x}{n})}).$$

Здесь и далее в этом подразделе $ln_k(x)$ означает $\underbrace{lnln\cdots ln}_{k}(x)$. Полученная оценка показывает, что мощность множества M_1 не больше остаточного члена в формулировке леммы.

Рассмотрим множество M_2 чисел $p \leq x$, для которых $\frac{p-1}{n}$ - простое и наибольший простой делитель числа $\frac{p-1}{n} - 1$ входит в разложение этого числа в степени не меньше 2. Тогда для $p \in M_2 \setminus M_1$ число $\frac{p-1}{n} - 1$ делится на квадрат числа, не меньшего, чем $y(f(x))$. Поэтому

$$\sharp M_2 \setminus M_1 \leq \sum_{m \geq y(f(x))} \frac{f(x)}{m^2} = \underline{O}(\frac{f(x)}{y(f(x))}).$$

Значит, мощность множества $M_2 \setminus M_1$ также не больше остаточного члена в формулировке леммы.

Для оставшихся простых p из множества, определённого в условии леммы 1, имеем:

$$\frac{p-1}{n} - 1 = p_k m^{(k-1)},$$

где p_k - простое число, $p_k > y(f(x))$, а $m^{(k-1)}$ — натуральное, такое, что $v(m^{(k-1)}) = k-1$. Кроме того, по построению, $m^{(k-1)} < \frac{f(x)}{y(f(x))}$. Количество указанных в условии простых (за исключением множеств M_1, M_2) не превосходит

$$\sum_{m \leq \frac{f(x)}{y(f(x))}, v(m)=k-1} N\left(p \leq x \;\middle|\; p, \frac{p-1}{n}, \frac{\frac{p-1}{n}-1}{m} - \text{простые}\right). \qquad (2.1)$$

Каждое слагаемое в этой сумме, соответствующее натуральному m, можно представить в виде

$$N\left(r \leq \left[\frac{\left[\frac{x-1}{n}\right]-1}{m}\right] \;\middle|\; r, rm+1, (rm+1)n+1 - \text{простые}\right). \qquad (2.2)$$

При $(m, n+1) \neq 1$ это 0. Будем считать, что $(m, n+1) = 1$.

Для дальнейших оценок используем следующую теорему из метода решета Сельберга.

Теорема 2.4. *([93, Теорема 2.4.6]) Пусть* $a_i, b_i \in \mathbb{Z}$, $i = 1, 2, \ldots, s$, $a_i \neq 0$, $(a_i, b_i) = 1$. *При* $i \neq k$, $(a_k, b_k) \notin \{(\pm a_i, b_i), (\pm a_i, -b_i)\}$. *Пусть* $w(p)$ - *число решений сравнения* $(a_1 u + b_1)(a_1 u + b_1) \cdots (a_s u + b_s) \equiv 0 \pmod{p}$, *а* $E = \prod_{1 \leq i \leq s} a_i \prod_{1 \leq i < k \leq s} (a_i b_k - a_k b_i)$.

Тогда при $N \geq 2$ *имеет место оценка*

$$N\,(u \leq N \,||a_i u + b_i| - \text{простые для всех } i = 1, \ldots, s) <$$

$$c(s) \frac{N}{ln^s N} \prod_{p|E} \left(1 - \frac{1}{p}\right)^{-(s - w(p))},$$

где постоянная $c(s)$ *зависит только от* s.

Применим эту теорему при $s = 3$, $(a_i, b_i) \in \{(1, 0), (m, 1), (mn, n+1)\}$. Тогда $E = m^2 n(n+1)(m(n+1) - mn) = m^3 n(n+1)$. Принимая во внимание, что

$$\left[\frac{\left[\frac{x-1}{n}\right] - 1}{m}\right] < \frac{x}{mn},$$

имеем следующую оценку выражения (2.2)

$$\leq C_1 \frac{x}{mnln^3 \frac{x}{mn}} \prod_{p|E} \left(1 - \frac{1}{p}\right)^{-(3 - w(p))},$$

где $w(p)$ - число решений сравнения $x(mx+1)(mnx+n+1) \equiv 0 \pmod{p}$. Если $p \mid m$, то $w(p) = 1$. Если $p \nmid m, p \mid n$, то $w(p) = 2$. Если $p \mid n+1$ (а значит, $p \nmid m, p \nmid n$), то $w(p) = 2$. Поэтому указанную оценку можно продолжить

$$\leq C_1 \frac{x}{mnln^3 \frac{x}{mn}} \prod_{p|m} \left(1 - \frac{1}{p}\right)^{-2} \prod_{p|n(n+1), p \nmid m} \left(1 - \frac{1}{p}\right)^{-1}$$

$$\leq C_1 \frac{xm^2n(n+1)}{mnln^3\frac{x}{mn}\varphi^2(m)\varphi(n(n+1))}.$$

Теперь всю сумму (2.1) можно оценить так:

$$\leq C_1 \frac{x(n+1)}{\varphi(n(n+1))} \sum_{m\leq \frac{f(x)}{y(f(x))}, v(m)=k-1} \frac{m}{\varphi^2(m)ln^3\frac{x}{mn}}.$$

Поскольку функция $\frac{z}{y(z)}$ возрастает, то эту оценку можно продолжить так:

$$< C_1 \frac{x(n+1)}{\varphi(n(n+1))} \sum_{m\leq \frac{x}{ny(\frac{x}{n})}, v(m)=k-1} \frac{m}{\varphi^2(m)ln^3\frac{x}{mn}}$$

$$< C_1 \frac{x(n+1)}{\varphi(n(n+1))ln^3(y(\frac{x}{n}))} \sum_{m\leq \frac{x}{ny(\frac{x}{n})}, v(m)=k-1} \frac{m}{\varphi^2(m)}$$

$$< C_1 \frac{x(n+1)(lnln\frac{x}{n})^3}{\varphi(n(n+1))ln^3(\frac{x}{n})} \frac{1}{(k-1)!} \left(\sum_{p<\frac{x}{ny(\frac{x}{n})}} \sum_{i=1}^{\infty} \frac{p^i}{\varphi^2(p^i)} \right)^{k-1}$$

$$= C_1 \frac{x(n+1)(lnln\frac{x}{n})^3}{\varphi(n(n+1))ln^3(\frac{x}{n})} \frac{1}{(k-1)!} \left(\sum_{p<\frac{x}{ny(\frac{x}{n})}} \left(\frac{p}{(p-1)^2} + \sum_{i=2}^{\infty} \frac{1}{p^{(i-2)}(p-1)^2} \right) \right)^{k-1}.$$

Так как

$$\sum_p \sum_{i=2}^{\infty} \frac{1}{p^{(i-2)}(p-1)^2} = \sum_p \frac{1}{(p-1)^2} \frac{1}{(1-\frac{1}{p})} < \infty,$$

а

$$\sum_{p<A} \frac{p}{(p-1)^2} = \sum_{p<A} \frac{1}{p} \left(1+\frac{1}{p-1}\right)^2 = \sum_{p<A} \frac{1}{p} \left(1+\frac{2}{p-1}+\frac{1}{(p-1)^2}\right) \leq lnlnA + C$$

для некоторой абсолютной константы C, то нашу оценку можно продолжить

$$\leq C_1 \frac{x(n+1)(lnln\frac{x}{n})^3}{\varphi(n(n+1))ln^3(\frac{x}{n})} \frac{1}{(k-1)!} \left(lnln\frac{x}{ny(\frac{x}{n})} + C_2 \right)^{k-1}$$

$$\leq C_1 \frac{x(n+1)(lnln\frac{x}{n})^3}{\varphi(n(n+1))ln^3(\frac{x}{n})} \frac{1}{(k-1)!} \left(lnln\frac{x}{n} + C_2\right)^{k-1}.$$

Лемма доказана.

Лемма 2.2. *(Эрдёш [93, гл. V §7]) Для любой положительной константы C и любого $\varepsilon > 0$ существует $\delta(\varepsilon) > 0$, такое, что*

$$\sum_{k\in\mathbb{N}, k\notin[(1-\varepsilon)lnlnz,(1+\varepsilon)lnlnz]} \frac{(lnlnz + C)^{k+2}}{(k-1)!} = \underline{O}(ln^{1-\delta(\varepsilon)}z)$$

при $z \to \infty$.

Применяя эту лемму с заменой z на $\frac{x}{n}$, из леммы 2.1 суммированием по k получим следующую лемму.

Лемма 2.3. *[29] Для любого $\varepsilon > 0$ существует $\delta(\varepsilon) > 0$, такое, что для любого $n \in \mathbb{N}, n \leq \frac{x}{y(x)}$ выполнено*

$$N\left(p \leq x \left| \frac{p-1}{n} = q \text{ - простое, } v(q-1) \notin [(1-\varepsilon)lnlnx, (1+\varepsilon)lnlnx]\right.\right)$$
$$= \overline{o}\left(\frac{x(n+1)}{\varphi(n(n+1))ln^{2+\delta(\varepsilon)}\frac{x}{n}}\right)$$

при $x \to \infty$, где $\overline{o}$ не зависит от n.

Доказательство теоремы 2.3. Для доказательства теоремы достаточно установить следующую оценку:

$$N\left(p \leq x \left| \begin{array}{r} \text{существует простое } q > y(x) \text{ такое, что } \frac{p-1}{q} = n \in \mathbb{N}, \text{ и} \\ v(q-1) \notin \left[(1-\varepsilon)lnlnx, (1+\varepsilon)lnlnx\right] \end{array}\right.\right) =$$
$$= \overline{o}\left(\frac{x}{ln^{1+\delta_1}x}\right),$$

где константа $\delta_1 > 0$.

Указанное количество можно оценить по лемме 2.3 суммированием по $n \leq \frac{x}{y(x)}$. Используя оценку Ландау для функции Эйлера, получим для некоторого $\delta > 0$ следующую оценку этой величины:

$$\overline{o}\left(\sum_{n\leq\frac{x}{y(x)}} \frac{x(n+1)}{\varphi(n(n+1))ln^{2+\delta}\frac{x}{n}}\right) = \overline{o}\left(xlnlnx\sum_{n\leq\frac{x}{y(x)}} \frac{1}{nln^{2+\delta}\frac{x}{n}}\right).$$

Нетрудно видеть, что при постоянном $s > 1$, так как $y(x) < x$, имеем

$$\sum_{2\leq n\leq\frac{x}{y(x)}} \frac{1}{nln^s\frac{x}{n}} \leq \int_1^{\frac{x}{y(x)}} \frac{dt}{tln^s\frac{x}{t}} \leq \int_{y(x)}^{x} \frac{zxdz}{xz^2ln^sz} = \int_{y(x)}^{x} \frac{dlnz}{ln^sz} = \underline{O}\left(\frac{1}{ln^{s-1}y(x)}\right).$$

Таким образом, рассматриваемое количество можно оценить так:

$$\overline{o}\left(xlnlnx\frac{(lnlnx)^{1+\delta}}{ln^{1+\delta}x}\right) = \overline{o}\left(\frac{x}{ln^{1+\delta_1}x}\right)$$

Теорема доказана.

Заметим, что теми же средствами могут быть получены аналогичные результаты для всех простых чисел из дерева числа n, отстоящих от n на постоянное, то есть не растущее с ростом n число звеньев. Для получения указанным способом таких оценок для остальных узлов дерева n необходимо в методе Сельберга получить более точную и эффективную зависимость от количества рассматриваемых линейных комбинаций. Практические выводы из результатов первой главы следующие:

1. Теорема о связи сложностей задач дискретного логарифмирования и Диффи-Хеллмэна и предпринятые исследования функции максимальной длины дерева Пратта свидетельствуют о надёжности протокола открытого распределения ключей Диффи-Хеллмэна при условии достаточной сложности задачи дискретного логарифмирования в той же группе.

2. Для стойкости задачи дискретного логарифмирования относительно предложенной в первой главе атаки с использованием маркеров следует выбирать такие простые значения p порядков используемых групп, для которых $p-1$ имеет "большой" простой делитель.

Для решения задачи дискретного логарифмирования в мультипликативной группе конечного поля имеется алгоритм с факторной базой. Этот алгоритм может быть распространён и на случай группы классов дивизоров гиперэллиптических кривых рода, большего, чем два [?]. Случай эллиптических кривых, род которых равен единице, на сегодняшний момент является наиболее привлекательным для построения систем защиты информации. В этом случае работают только алгоритмы общего применения, наилучшим из которых является сейчас алгоритм Полларда с оценкой сложности порядка корня из числа элементов рассматриваемой группы.

Задача открытого распределения ключа возникает при инициации сеанса защищённой связи. Если задача дискретного логарифмирования, привлекающая к своему решению большое количество математиков, окажется дискредитированной — хотелось бы иметь ей замену. Построению такой замены на основе задачи факторизации посвящена вторая глава.

Глава 3

Использование некоммутативной операции для построения систем защиты информации

Большинство известных в настоящее время схем открытого распределения ключей основаны на стойкости задачи дискретного логарифмирования. Эта задача давно притягивает большое количество исследователей и имеет для своего решения множество алгоритмов, среди которых пока нет полиномиального. Построение эффективной замены дискретного логарифмирования в схеме открытого расключей — очень важная задача при решении проблемы построения надёжных систем защиты информации. В 1993г. В. М. Сидельниковым в совместной работе с автором диссертации была высказана альтернативная идея: использовать некоммутативную операцию. Однако конкретных примеров операций, способных обеспечить работу и стойкость таких схем открытого распределения ключа, хотя бы относительно простейших подходов, пока найдено не было. В этом разделе приведён криптоанализ и предложены некоторые примеры таких операций.

В главе обсуждается предложенный В.М.Сидельниковым способ выработки "открытого"ключа, основанный на использовании свойств некоммутативных групп. Рассмотрены четыре варианта [35]: матричные группы, эллиптические

модули, кольцо, полученное факторизацией кольца автоморфизмов некоторого конечного поля, а также операция с использованием символа степенного вычета. Оценена сложность вскрытия получаемых при этом схем и показано, что стойкость первых трёх из них невысока. Затем предложена новая операция (вариант 4) [37]. Атаки, оказавшиеся эффективными в предыдущих случаях для этой операции невозможны.

Описание системы предложенной в [35].

В открытом доступе имеется описание некоторой группы G и двух ее коммутативных подгрупп H и R, при этом предполагается, что не единичные элементы $h \in H, r \in R$ не коммутируют. Абоненты A и B для выработки общего секретного ключа поступают следующим образом. Каждый из них независимо друг от друга случайно вырабатывает по одному элементу из H и R и в последующем держит их в секрете: $h_A \in H, r_A \in R$ для A, $h_B \in H, r_B \in R$ для B. Далее они находят элементы $g_A = h_A r_A$, $g_B = h_B r_B$ и обмениваются ими по открытому каналу связи (и, тем самым, делают их общедоступными). Для выработки общего ключа абонент A берет свои "секретные"элементы h_A, r_A и полученный по открытому каналу связи элемент g_B и образует произведение $h_A g_B r_A$. Аналогично абонент B образует произведение $h_B g_A r_B$. Из определений вытекает, что

$$h_A g_B r_A = h_A(h_B r_B) r_A = (h_A h_B)(r_B r_A) = (h_B h_A)(r_A r_B) =$$
$$h_B(h_A r_A) r_B = h_B g_A r_B = g$$

Тем самым, общий секретный ключ g выработан.

Заметим, что если r_A коммутирует с h_B (r_B коммутирует с h_A), то $g = g_A g_B$ ($g = g_B g_A$) и, тем самым, не является секретным. Если внешний наблюдатель хочет найти секретный ключ g, то он должен решить следующую задачу: по известным описаниям G, H, R и известным элементам $g_A \in G$, $g_B \in G$ найти элемент $g = h_A g_B r_A = h_B g_A r_B$, не зная элементов h_A, r_A, h_B, r_B. Естественный

путь для решения этой задачи: сначала "разложить на множители"известный элемент $g_B \in G$, (или $g_A \in G$), то есть найти его представление в виде

$$g_B = h_B r_B, h_B \in H, r_B \in R$$

с неизвестными h_B, r_B. Такое представление единственно из-за условия некоммутативности элементов из H и R. Действительно, если бы существовало два различных представления

$$h_B r_B = \tilde{h}_B \tilde{r}_B,$$

то тогда неединичный элемент $\tilde{h}_B^{-1}\tilde{h}_B = \tilde{r}_B {r_B}^{-1}$ лежал бы в пересечении H и R, тем самым, коммутировал бы и с H и с R. Пусть эта задача решена. Тогда внешний наблюдатель берет найденные им элементы $h_B \in H, r_B \in R$, известный элемент $g_A \in G$ и находит интересующий его ключ $h_B g_A r_B$.

Таким образом, стойкость описанной системы определяется сложностью "разложения на множители"в группе G, то есть решением заведомо совместного уравнения

$$g = hr, g \in G, h \in H, r \in R$$

с известным g и неизвестными h и r. Иногда удобнее это уравнение записывать в следующем виде:

$$y = gx, g \in G, y \in H, x \in R \tag{3.1}$$

с известным g и неизвестными y и x. При этом сложность решения меняется несущественно: добавляется обращение элемента r - нахождение r^{-1}, что при знании порядка группы является полиномиально разрешимой задачей.

Резюмируем для удобства все требования на группы, вытекающие из соображений реализации системы и оценки ее стойкости:

1. Свойства групп H, R позволяют эффективно вырабатывать случайные элементы из H и R.

2. Свойства групп G, H, R позволяют эффективно вычислять произведения элементов из этих групп.

3. Группа G такова, что позволяет эффективно передавать по открытому каналу связи ее элементы.

4. Сложность решения заведомо совместного уравнения (3.1) достаточно высока.

Подчеркнем, что требование "любые не единичные элементы $h \in H, r \in R$ не коммутируют"не включено в этот список. Дело в том, что это требование очень сильное и существенно ограничивает класс допустимых групп. Оно введено выше только для упрощения первого описания системы, поскольку гарантирует невозможность явного вы-ражения секретного элемента g через известные элементы g_A и g_B (в виде $g = g_A g_B$, или $g = g_B g_A$). В общем случае, когда не исключена вероятность выбора коммутирующих элементов r_A и h_B , r_B и h_A, при анализе и оценке стойкости системы необходимы дополнительные рассмотрения. При этом стойкость системы понижается из-за ненулевой вероятности появления таких ситуаций.

Одновременно с этим пропадает свойство единственности разложения на множители, единственности решения уравнения (3.1). Но это не меняет логики описанного метода нахождения секретного ключа g: для этого достаточно использовать произвольное разложение на множители". Оба разных разложения дают один и тот же элемент g. Действительно, пусть

$$g_B = h_B r_B = \tilde{h}_B \tilde{r}_B$$

с некоторыми $h_B, \tilde{h}_B \in H, r_B, \tilde{r}_B \in R$. Тогда

$$c = \tilde{h}_B^{-1} \tilde{h}_B = \tilde{r}_B \tilde{r}_B^{-1} \in H \cap R,$$

и элемент c коммутирует со всеми элементами из H и R. Тогда

$$\tilde{h}_B g_A \tilde{r}_B = (\tilde{h}_B c) g_A (c^{-1} \tilde{r}_B) = h_B g_A r_B = g.$$

С учетом этих замечаний дальше не будем следить за выполнением требования о том, чтобы любые не единичные элементы $h \in H, r \in R$ не коммутировали, а для уравнения (3.1) будем искать произвольное решение.

Ниже будут рассмотрены некоторые алгебраические структуры с целью нахождения групп с высокой сложностью решения уравнения (3.1) с учетом требований 1-3.

Общие замечания о решении уравнения (3.1).

1) Очевидно, что сложность решения уравнения (3.1) в группе $\tilde{G}$, гомоморфном образе группы $G, \varphi : G \to \tilde{G}(\tilde{H} = \varphi(H), \tilde{R} = \varphi(R))$, в случае если изоморфизм $\tilde{G} \cong G/Ker\varphi$ установлен не выше сложности решения этого уравнения в группе G. В самом деле, если требуется решить уравнение

$$\tilde{y} = \tilde{g}\tilde{x}$$

с известным $\tilde{g}$ и неизвестными $\tilde{y} \in \tilde{H}, \tilde{x} \in \tilde{R}$, то вначале находим какой-нибудь элемент $g \in G$ такой, что

$$\varphi(g) = \tilde{g}.$$

Затем решаем уравнение

$$y = gx$$

в группе G с найденным $g \in G$ и неизвестными $y \in H, x \in R$. Это решает также и сходное уравнение, поскольку

$$\varphi(y) = \varphi(g)\varphi(x) = \tilde{g}\varphi(x).$$

2) Если группа G содержит нетривиальную нормальную подгруппу G_1 , то при естественном гомоморфизме $G \to G/G_1 = G_\varphi$ группы H и R отобразятся в некоторые коммутативные подгруппы, соответственно $H_\varphi \cong H/H \cap G_1$ и $R_\varphi \cong R/R \cap G_1$ группы G_φ. Уравнение (3.1) при этом перейдет в уравнение

$$y_\varphi = g_\varphi x_\varphi, g_\varphi \in G_\varphi \tag{3.2}$$

с неизвестными $y_\varphi \in H_\varphi$, и $x_\varphi \in R_\varphi$.

Если обозначить $\mid G \mid$ порядок группы G, то очевидно, что

$$\mid G_\varphi \mid = \mid G \mid / \mid G_1 \mid, \mid H_\varphi \mid = \mid H \mid / \mid H \cap G_1 \mid, \mid R_\varphi \mid = \mid R \mid / \mid R \cap G_1 \mid .$$

Поэтому если группа G_1 достаточно велика, то уравнение (3.2) может решаться проще (круг поиска сужается), чем уравнение (3.1). После того, как уравнение (3.2) будет решено, решения уравнения (3.1) можно искать в соответствующих смежных классах.

Таким образом, возникает естественное требование, чтобы группа G не содержала нетривиальных нормальных подгрупп, то есть была простой. Целесообразность этого требования подтверждается еще и тем обстоятельством, что при его выполнении имеем:

$$G' = G; G^k = G, k \in \mathbb{N}; C(G) = e,$$

где G' - коммутатор, а $C(G)$ - центр группы G, так как все группы, стоящие в левых частях этих равенств, являются нормальными подгруппами группы G.

Заметим, что в наших рассуждениях используются только те элементы из группы G, которые являются всевозможными конечными произведениями элементов групп H и R. Поэтому условие простоты более естественно накладывать на подгруппу P, порожденную этими произ-ведениями (если K - нормальная подгруппа в G, то $K \cap P$ - нормальная подгруппа в P).

Полезным приемом решения уравнения (3.1) является переход к коммутативным расширениям подгрупп H и R. Если в группе G существуют такие коммутативные подгруппы H^e и R^e, что

$$H \subset H^e, R \subset R^e,$$

то разложить на множители, или решить уравнение (3.1) в неизвестных $y \in H^e$ и $x \in R^e$, может быть проще, чем в неизвестных $y \in H$ и $x \in R$ (расширяется количество решений и может упроститься структура подгруппы). При этом оказывается, что секретный элемент g может быть выражен также и через некоторые множители из H^e и R^e, а не только из H и R. В самом деле, пусть

$$g_B = h_B r_B = \tilde{h}_B \tilde{r}_B, h_B \in H, r_B \in R, \tilde{h}_B \in H^e, \tilde{r}_B \in R^e,$$

тогда

$$\tilde{h}_B^{-1} {h_B}^{-1} = \tilde{r}_B^{-1} {r_B}^{-1} = c \in H^e \cap R^e,$$

где элемент с коммутирует со всеми элементами из H^e и R^e. Поэтому

$$\tilde{h}_B g_A \tilde{r}_B = (\tilde{h}_B c) g_A (c^{-1} \tilde{r}_B) = h_B g_A r_B = g.$$

Легко видеть, что можно ослабить требование на H и R. Достаточно требовать выполнения следующего условия: $H^e \supset H, R^e \supset R$ - такие расширения, что любой элемент $c \in H^e \cap R^e$ коммутирует со всеми элементами из H и R. Тогда разложение известного элемента g_B на множители из H^e и R^e приводит тем же методом к нахождению секретного элемента g.

Таким образом, на подгруппы тоже можно наложить естественное требование, чтобы их нельзя было расширить с сохранением коммутативности.

В заключение заметим, что если отказаться от условия существования

обратного элемента (при этом все рассмотренные выше группы становятся полугруппами), то решение уравнения (3.1), или уравнения $xy = g$, не приводит в общем случае к нахождению ключа.

Для начала было предпринято исследование хорошо известных некоммутативных групп с целью выявить возможность их применения в рассматриваемой схеме открытого распределения ключей.

Вариант 1. Группа матриц.

Пусть G - группа невырожденных матриц размера $n \times n$ над некоторым полем $\mathbb{K}, R = \langle A \rangle, H = \langle B \rangle$ - циклические подгруппы порожденные невырожденными матрицами из G: A и B соответственно. Уравнение (3.1) будем решать в следующих расширениях подгрупп H и R: R^e - все невырожденные матрицы вида $\sum_{i=0}^{n-1} x_i A^i, x_i \in \mathbb{C}, i = 0, 1, ..., n-1$, и обратные к ним; H^e - все невырожденные матрицы вида $\sum_{i=0}^{n-1} y_i B^i, y_i \in \mathbb{C}, i = 0, 1, ..., n-1$ и обратные к ним. При этом уравнение (3.1) может быть записано в виде

$$\sum_{i=0}^{n-1} y_i B^i = D(\sum_{i=0}^{n-1} x_i A^i) \tag{3.3}$$

с известной матрицей D вида $B^b A^a; b, a \in \mathbb{Z}$ и неизвестными $x_i, y_i \in \mathbb{K}, i = 0, 1, ..., n-1, j = 0, 1, ..., n-1$.

Известно, что матрица является корнем своего характеристического многочлена

$$\chi(t) = det(A - tE) = \sum_{i=0}^{n} a_i^0 t^i, a_i{}^0 \in \mathbb{C}, a_0^0 = detA \neq 0, a_n^0 = 1,$$

то есть

$$\chi(A) = \sum_{i=0}^{n} a_i^0 A^i = 0.$$

Кроме того, для матрицы

$$A' = -\frac{1}{a_0{}^0}\sum_{i=0}^{n-1} a_{i+1}^0 A^i$$

имеем

$$A'A = -\frac{1}{a_0^0}\sum_{i=1}^{n} a_i^0 A^i = -\frac{1}{a_0^0}(\chi(A) - a_0^0 E) = E,$$

откуда $A' = A^{-1}$.

Поэтому любая целая степень матрицы A представляется как линейная комбинация матриц $E, A, \ldots, A^{n-1}$ над $\mathbb{K}$. Поэтому матричное уравненние (3.3) заведомо разрешимо.

Матричное уравнение (3.3) в скалярной записи является системой из n^2 линейных однородных уравнений в поле $\mathbb{K}$ относительно $2n$ неизвестных x_i, y_j, связанных условием, что матрицы $\sum_{i=0}^{n-1} x_i A^i, \sum_{j=0}^{n-1} y_j B^j$ - невырождены. Хорошо известно, что эти условия эквивалентны следующим:

$$\sum_{i=0}^{n-1} x_i \lambda^i \neq 0 \quad \text{для любого } \lambda \text{ - собственного значения матрицы } A, \tag{3.4}$$

$$\sum_{j=0}^{n-1} y_j \mu^j \neq 0 \quad \text{для любого } \mu \text{ - собственного значения матрицы } B. \tag{3.5}$$

Итак, в рассматриваемом случае уравнение (3.1) сведено к системе из n^2 однородных уравнений (3.3) и $k+1$ неравенств (3.4), (3.5), где k, l - число различных собственных значений матриц A и B соответственно. Очевидно, что число операций для решения этой задачи не превосходит $O(n^3)$, поскольку для нахождения базиса решений системы (3.3) требуется не более чем $(2n)^3$ операций, а нахождение решения, удовлетворяющего условиям (3.4),(3.5) производится путем проверки этих условий на этом базисе.

Вариант 2. Эллиптические модули.

Введем сначала необходимые понятия и определения. Пусть $\mathbb{F}_q, q = p^m, m \in \mathbb{N}$, - конечное поле из q элементов. Рассмотрим алгебраическим расширением степени n поля $\mathbb{F}_q$ в виде $A = F_q[T]/(f)$, где T - независимая переменная, а $f \in \mathbb{F}_q[T]$ - неприводимый многочлен степени n по T.

Поле A обладает следующими эндоморфизмами $: x \to ax$, где $a \in A$, и $\tau\colon x \to x^q$ (эндоморфизм Фробениуса). Поэтому и любой формально составленный многочлен с коэффициентами из A от переменной τ может быть интерпретирован как эндоморфизм поля A. Множество всех таких многочленов $A[\tau]$ является кольцом относительно композиции эндоморфизмов, в котором многочлены $a, \tau \in A[\tau]$ перемножаются с учетом правила $\tau a = a^q \tau$, что позволяет найти и произведение любых двух многочленов. Кольцо $A[\tau]$ является некоммутативным кольцом.

Очевидно, τ является линейным над $\mathbb{F}_q$ оператором, действующим в поле A : $\tau(\alpha a + \beta b) = \alpha\tau(a) + \beta\tau(b), \alpha, \beta \in \mathbb{F}_q$. Следовательно $A[\tau]$ можно рассматривать как кольцо линейных операторов на A.

Определение. *(см. [121]) Эллиптическим модулем ранга r называется гомоморфизм φ кольца A в кольцо $A[\tau]$ такой, что $\varphi(b) = b$ при любом $b \in \mathbb{F}_q$, и степень многочлена $\varphi(T)$ по переменной τ равна r.*

Отметим, что для корректности этого определения необходимо, чтобы образ $\varphi(f(T))$ неприводимого многочлена $f(T)$ определяющего поле A был нулевым оператором (например, многочленом из кольца $A[\tau]$ кратным характеристическому многочлену оператора τ). Очевидно, что эллиптический модуль полностью определяется заданием элемента $\varphi(T) \in A[\tau]$.

Очевидно также, что для любого $a \in A$ выполнено $deg_T\varphi(a) \leq n - 1, deg_\tau\varphi(a) \leq r(n-1)$. Для построения системы с открытым ключом на основе эллиптических модулей положим $G = A[\tau]$, рассматриваемом как кольцо линейных операторов на A, $H = \varphi_\beta(A), R = \varphi_\alpha(A)$ для некоторых различных эллиптических модулей $\varphi_\alpha, \varphi_\beta$. A - поле, а гомоморфизм переводит обратный

элемент в обратный, поэтому H и R являются коммутативными группами.

Система, построенная таким образом, кажется предпочтительнее предыдущей, ввиду того, что подгруппы H и R не являются в ней циклическими, как это было в предыдущем случае. Однако их строение также достаточно просто, что позволяет легко решать уравнение (3.1).

Так как гомоморфизм φ_α сохраняет операцию сложения и оставляет на месте элементы из $\mathbb{F}_q$, а поле A является линейным пространством над $\mathbb{F}_q$ с базисом $1, T, \ldots, T^{n-1}$, то множество $\varphi_\alpha(A) = \langle 1, \varphi_\alpha(T), \ldots, (\varphi_\alpha(T))^{n-1} \rangle_{\mathbb{F}_q}$ является линейным пространством над $\mathbb{F}_q$. Тоже самое, очевидно, справедливо и для гомоморфизма φ_β.

Таким образом, для любых $a, b \in A$

$$\varphi_\alpha(a) = \sum_{i=0}^{n-1} x_i (\varphi_\alpha(T))^i,$$

$$\varphi_\beta(b) = \sum_{i=0}^{n-1} y_i (\varphi_\beta(T))^i,$$

где $x_i, y_i \in \mathbb{F}_q, i = 0, 1, \ldots, n-1$ и уравнение (3.1) записывается в виде

$$(\sum_{i=0}^{n-1} y_i (\varphi_\beta(T))^i) = D(\sum_{i=0}^{n-1} x_i (\varphi_\alpha(T))^i), \tag{3.6}$$

где элементы

$$(\varphi_\alpha(T))^i, (\varphi_\beta(T))^i, i = 0, 1, \ldots, n-1, D = \varphi_\beta(a)\varphi_\alpha(b) \in A[\tau] \tag{3.7}$$

известны, а $x_i, y_i \in \mathbb{F}_q$ - неизвестны.

Элемент τ является линейным оператором на n мерном векторном пространстве A, и, следовательно, может быть записан в виде матрицы размера $n \times n$ с коэффициентами из $\mathbb{F}_q$. Тоже самое выполнено и для всех элементов (3.7). Причем, как было показано выше, все эти линейные операторы, а также операторы, полученные нетривиальными линейными комбинациями над $\mathbb{F}_q$

операторов $(\varphi_\alpha(T))^i, i = 0, 1, \ldots, n-1$, или операторов $(\varphi_\beta(T))^i, i = 0, 1, \ldots, n-1$, являются невырожденными.

Таким образом, равенство (3.6) можно представить как заведомо разрешимую систему из n^2 линейных однородных уравнений от $2n$ неизвестных $x_i, y_i, i = 0, 1, \ldots, n-1$.

Отметим, что n должно быть ограничено, ввиду того, что передаваемые по каналам связи элементы вида $\varphi_\beta(a)\varphi_\beta(b)$ являются многочленами степени $n-1$ по T и $(r_\alpha + r_\beta)(n-1)$ по τ, где r_α и r_β - ранги эллиптических модулей соответственно φ_α и φ_β, и по этим же причинам $(r_\alpha + r_\beta)(n-1)$ не может быть слишком большим.

Вариант 3. Кольцо $K = A[\tau]/\tau^t$.

Рассмотрим кольцо $K = A[\tau]/\tau^t$, где $A[\tau]$ - кольцо многочленов от τ, взятое из предыдущего пункта, а $t = 2k, k \in \mathbb{N}\backslash\{1\}, t > n$.

Элементами кольца K являются смежные классы вида

$$\overline{x}_0 = x + f\tau^t,$$

где f пробегает элементы, принадлежащие $A[\tau]$.

Определим операцию умножения двух смежных классов $\overline{x}_1$ и $\overline{x}_2$ следующим образом:

$$\overline{x}_1 * \overline{x}_2 = \overline{x_1 x_2}. \tag{3.8}$$

Для любого многочлена

$$f = \sum_{i=0}^{t-1} a_i \tau^i \in A[\tau]$$

имеем

$$\tau^t f = (\sum_{i=0}^{t-1} a_i{}^q \tau^i)\tau^t = \tilde{f}\tau^t,$$

поэтому для любых $f_1, f_2 \in A[\tau]$ имеем:

$$(x_1 + f_1\tau^t)(x_2 + f_2\tau^t) = x_1x_2 + f_1\tau^t x_2 + x_1 f_2\tau^t + f_1\tau^t f_1\tau^t = x_1x_2 + f_3\tau^t, f_3 \in A[\tau],$$

и, тем самым, операция умножения (3.8) в кольце K определена корректно. Кроме того, операция умножения в кольце $A[\tau]$ ассоциативна, поэтому и операция (3.8) также ассоциативна.

Нетрудно видеть, что нильпотентными элементами кольца K являются те и только те многочлены, которые не содержат свободного члена.

Докажем, что любой элемент кольца K, отличный от нильпотентного, имеет обратный элемент.

Пусть

$$f = \sum_{i=0}^{t-1} a_i \tau^i, a_i \in A, a_0 \neq 0.$$

Покажем, что существует многочлен

$$f_1 = \sum_{i=0}^{t-1} x_i \tau^i, x_i \in A,$$

такой, что

$$\overline{f}_1 * \overline{f} = \overline{g}, \tag{3.9}$$

где g - произвольный элемент поля A. После подстановки выражений для f_1 и f и раскрытия скобок это уравнение становится эквивалентным системе из $t-1$ линейного однородного уравнения с коэффициентами из поля A относительно t неизвестных $x_i \in A, i = 0, 1, \ldots, t-1$. Пусть $x_i^0, i = 0, 1, \ldots, t-1$ - какое-либо нетривиальное решение этой системы, а i_0 - минимально возможный индекс для которого $x_{i_0}^0 \neq 0$. Тогда левая часть уравнения (3.9) содержит элемент $a_0 x_{i_0}^0 \tau^{i_0}$, где $a_0 x_{i_0}^0 \in \backslash\{0\}$, откуда $i_0 = 0$ и $a_0 x_0^0 = g \neq 0$.

Таким образом, для многочлена

$$f_2 = \sum_{i=0}^{t-1} x_i^0 g^{-1} \tau^i, x_0^0 g^{-1} \in A\{0\},$$

$$\overline{f}_2 * \overline{f} = \overline{g}^{-1} * \overline{f}_1 * \overline{f} = \overline{g} * \overline{g}^{-1} = \overline{gg^{-1}} = e^{-1},$$

где e - единица поля $\mathbb{F}_q$. Мы доказали, что любой элемент из кольца K, отличный от нильпотентного, обладает левым обратным элементом, который, в свою очередь, тоже не является нильпотентным элементом. Дальнейшие рассуждения (см., например, [117] стр. 31) показывают, что левый обратный в этом случае совпадает с правым обратным элементом.

Таким образом, множество многочленов кольца K, имеющих ненулевой свободный член, образует некоммутативную группу K^*. Возьмем в качестве G группу K^*. Получим конечную некоммутативную группу, элементами которой являются многочлены с коэффициентами из $\mathbb{F}_q$ от двух переменных T и τ степени $n-1$ по первой и $t-1$ по второй.

Однако указанная схема не является стойкой. Нетрудно понять, что для вскрытия рассматриваемой схемы открытого распределения ключа достаточно решить уравнение

$$(\sum_{i=0}^{t-1} x_i\tau^i)(\sum_{i=0}^{t-1} a_i\tau^i) = \sum_{i=0}^{t-1} y_i\tau^i, \tag{3.10}$$

или

$$(\sum_{i=0}^{t-1}(\sum_{k+l=i}^{t-1} x_k a_l^{p^k})\tau^i) = \sum_{i=0}^{t-1} y_i\tau^i, \tag{3.11}$$

или

$$\sum_{k+l=i}^{t-1} x_k a_l^{p^k} = y_i, i = 0, 1, \ldots, t-1, \tag{3.12}$$

где коэффициенты $a_i \in \mathbb{F}_q$ известны, а x_i, y_i неизвестны и должны быть найдены такими, чтобы соответствующие суммы принадлежали определённым заранее коммутативным подгруппам. Поскольку уравнения (3.12) линейны относительно координат x_i, y_i и все их решения могут быть выписаны в виде линейных выражений от неопределённых коэффициентов, то сложность должна содержаться в том, чтобы удовлетворить этим условиям принадлежности. Добиться этого пока не удалось.

Вариант 4. Символ степенного вычета.

Пусть имеется некоторая некоммутативная полугруппа $(G, *)$ с полиномиально вычислимой, ассоциативной операцией $*$ и двумя коммутативными подполугруппами G_1 , G_2 , причём элементы из разных подполугрупп между собой не коммутируют.

Ещё раз напомним схему. Абоненты A и B случайным, независимым друг от друга образом вырабатывают секретные элементы $a_1 \in G_1, a_2 \in G_2$ и $b_1 \in G_1, b_2 \in G_2$ соответственно (секретные элементы будем в дальнейшем обозначать жирными буквами). Затем они передают друг другу по открытому каналу связи произведения $a_1 * a_2$ и $b_1 * b_2$. Общий секретный ключ абоненты и вычисляют соответственно по формулам $k = a_1 * (b_1 * b_2) * a_2$ и $k = b_1 * (a_1 * a_2) * b_2$.

Пример 1. Пусть $a * b = ab \left(\frac{\eta(|a|)}{\mu(|b|)}\right)_p$, где целые a, b состоят из достаточно больших простых чисел (чтобы нельзя было быстро разложить на множители), η, μ - мультипликативные функции натурального аргумента, $|a|$ - модуль числа a, а скобки обозначают символ Якоби. Ясно, что эта операция может быть выполнена за полиномиальное время, и легко проверить, что она удовлетворяет условию ассоциативности:

$$a * (b * c) = \left(\frac{\eta(|a|)}{\mu(|bc|)}\right) a \left(\frac{\eta(|b|)}{\mu(|c|)}\right) bc = \left(\frac{\eta(|a|)}{\mu(|b|)}\right) ab \left(\frac{\eta(|ab|)}{\mu(|c|)}\right) c = (a * b) * c$$

Ясно, что секретная часть общего ключа k - это знак. Однако, и он в данном случае может быть вычислен благодаря следующим двум свойствам рассматриваемой операции:

$$a * b \neq b * a \text{следовательно} a * b = -b * a, \tag{3.13}$$

$$a * (-b) = -a * b \tag{3.14}$$

Действительно,

$$a_1 * a_2 * [-(b_1 * b_2)] = a_1 * b_1 * a_2 * b_2 = k.$$

Этот недостаток можно преодолевать изменением структуры подполугрупп G_1, G_2 так, чтобы при $a \in G_1$ и $b \in G_2$ по a, b нельзя было за полиномиальное время вычислить знак в равенстве

$$a * b = \pm b * a.$$

Однако мы поступим иначе. Введение обобщенного символа Якоби позволяет увеличить объём секретной части.

Обозначим для некоторого простого p : $\xi = exp(2\pi i/p)$. В кольце целых алгебраических чисел $K = \langle 1, \xi, ..., \xi^{p-2} \rangle_Z$ кругового поля $\mathbb{Q}(\xi)$ обозначим $\nu_J()$ - продолжение на $\mathbb{Q}(\xi)$ J-адического нормирования для простого идеала J кольца K, $\nu_p()$ - p-адическое нормирование поля рациональных чисел $\mathbb{Q}$, а $N(J) = \sharp(K/J)$ норма идеала J. Пусть $\alpha, \gamma \in \mathbb{Q}(\xi), \nu_\lambda(\gamma) = 0$ (то есть λ не делит ν) , где $\lambda = 1 - \xi$. Обобщённый символ Якоби степени p [122] определяется как произведение по всем простым идеалам:

$$\left(\frac{\alpha}{\gamma}\right)_p = \prod_{J \nmid p, \alpha} \left(\frac{\alpha}{J}\right)_p^{\nu_J(\gamma)}, \quad \text{где} \quad \left(\frac{\alpha}{J}\right)_p = \xi^j \equiv \alpha^{\frac{N(J)-1}{p}} (mod J)$$

В частности, при $p = 2$ это обычный символ Якоби и Лежандра.

Пример 2. Пусть

$$a * b = ab \left(\frac{\eta(a)}{\mu(b)}\right)_p, \tag{3.15}$$

где $a, b \in \mathbb{Q}(\xi) \backslash \{0\}$, а η, μ - мультипликативные функции на $Q(\xi) \backslash \{0\}$, такие, что $\eta(\xi), \mu(\xi) = 1$. Ассоциативность этой операции доказывается аналогично примеру 1. Свойство (3.13) для этой операции не выполняется, а свойство (3.14) приобретает вид

$$a * (\xi^t b) = \xi^t a * b, \quad \text{для любого} \quad t \in \mathbb{N}. \tag{3.16}$$

Свойства обобщённого символа Якоби хорошо известны (см. [122]). В частности, как видно уже из определения, выполнено свойство мультипликативности по верхнему и нижнему индексу, кроме того, верхний индекс можно брать по модулю нижнего. Выполнен так же и закон взаимности, установленный ещё Куммером, а именно:

Теорема 3.1. *(см.[122] стр.172) Пусть*

$$\alpha = 1 + \lambda^2 u, u \in \mathbb{Q}(\xi), \nu_\lambda(u) \geq 0,$$

$$\beta = 1 + \lambda v, v \in \mathbb{Q}(\xi), \nu_\lambda(v) \geq 0.$$

Тогда

$$\left(\frac{\alpha}{\beta}\right)_p \left(\frac{\beta}{\alpha}\right)_p^{-1} = \xi^{\frac{T(\xi log\alpha Dlog\beta)}{p}},$$

$$\left(\frac{\xi}{\alpha}\right)_p = \xi^{\frac{T(log\alpha)}{p}},$$

где T - след $\mathbb{Q}_p(\xi)$ над $\mathbb{Q}_p$ ($\mathbb{Q}_p$ - p - адическое замыкание поля рациональных чисел),

$$log\alpha = log(1 + \lambda^2 u) = \lambda^2 u - (\lambda^2 u)^2/2 + (\lambda^2 u)^3/3 - ... \in \mathbb{Q}_p(\xi),$$

$$Dlog\beta = \beta'/\beta \in \mathbb{Q}_p(\xi),$$

где β' - формальная производная по λ.

Следствие 3.1. *(см.[123] стр.181) Если $p \geq 3$, то*

$$\alpha, \beta \in \{1 + x\lambda^{(p+1)/2} \mid x \in \mathbb{Q}(\xi), \nu_\lambda(x) \geq 0\} \Rightarrow \left(\frac{\alpha}{\beta}\right)_p = \left(\frac{\beta}{\alpha}\right)_p.$$

Из этих утверждений следует, что если η, μ принимают значения из $\mathbb{Q}(\xi)$, то относительно операции (3.15) $K\backslash 0$ является, вообще говоря, некоммутативной полугруппой, где 1 — двусторонняя единица.

Функции η, μ могут быть устроены, например, следующим образом: $f(N(a))$, где $N(a) \in \mathbb{Z}$, а f - мультипликативная функция целого аргумента, заданная на простых рациональных числах при помощи равенства

$$f(q) = \prod \alpha_q,$$

где справа стоит единица или произведение по некоторому конечному набору элементов из $\mathbb{Q}(\xi)$, зависящих от q и имеющих вид

$$1 \pm \lambda^{\alpha}, \alpha \in \mathbb{N}. \tag{3.17}$$

Рассмотрим символ Якоби со взаимно простыми верхним и нижним индексом такого вида. Допустим также, что показатели степени в выражении (3.17) для верхнего и нижнего индекса взаимно просты. Такой символ может быть вычислен не более чем за $(log(max\{\alpha, \beta\}))$ переворотов и взятий верхнего индекса по модулю нижнего. Действительно, эти преобразования не меняют структуру верхнего и нижнего индекса и они остаются взаимно простыми, а взятие по модулю сводится к делению одного показателя на другой с остатком. В этой связи, в силу взаимной простоты показателей, всё сведётся к вычислению символа вида

$$\left(\frac{2}{1-\lambda}\right)_p = \left(\frac{2}{\xi}\right)_p = 1$$

или

$$\left(\frac{2}{1+\lambda}\right)_p = \left(\frac{2}{2+\xi}\right)_p,$$

который после одного переворота можно посчитать по определению, используя известные выражения для нормы и количества простых идеалов - делителей числа 2 (см. например [123] стр. 179).

Один переворот осуществляется за полиномиальное от p время. Действительно, для отображения следа $T : K \to \mathbb{Z}$ при $0 < i < p$, имеем

$$T(\xi^j) = \sum_{j=1}^{p-1} (\xi^j)^i = \sum_{t=1}^{p-1} \xi^t = -1, \quad T(1) = \sum_{j=1}^{p-1} 1 = p - 1,$$

$$T(\lambda^i) = T((1-\xi)^i) = T(\sum_{k=0}^{i} C_i^k \xi^k (-1)^k) = \sum_{k=0}^{i} C_i^k T(\xi^k)(-1)^k =$$

$$= p - \sum_{k=0}^{i} C_i^k (-1)^k = p.$$

Поскольку $p = \lambda^{p-1} u$, где $u \in \mathbb{K}$ - обратимый элемент, то при $i \geq p$ для $\lambda^{i-p} u = a_0 + a_1 + ... + a_{p-2}\lambda^{p-2} \in K, a_i \in \mathbb{Z}$, получим

$$T(\lambda^i) = pT(\lambda^{i-p+1} u) = pT(a_0\lambda + a_1\lambda^2 + ... + a_{p-2}\lambda^{p-1}) =$$

$$p(a_0 T(\lambda) + ... + a_{p-2} T(\lambda^{p-1})) = p^2 q_i, q_i \in \mathbb{Z}.$$

Поэтому $p^2 \mid T(\lambda^i)$ при $i \geq p$, то есть формула для вычисления закона взаимности не зависит от степеней выше, чем $p - 1$.

Таким образом, при небольших значениях p операция (3.15) эффективно вычисляется, в то время как соответствующее разложение методом полного перебора возможных сомножителей потребует экспоненциального по p времени (каждый сомножитель может, вообще говоря, иметь p коэффициентов). Коммутирующие множества могут быть построены при помощи указанного выше следствия или свойства (3.16). Пусть

$$a * b = \xi^t b * a, a * c = \xi^s c * a$$

Тогда для $d = \underbrace{b * ... * b}\, x * \underbrace{c * ... * c}\, y$

$$a * d = d * a \Leftrightarrow xt + ys \equiv 0 (\mathrm{mod} p).$$

Последнее сравнение при любых фиксированных s, t даёт как минимум p различных по модулю p пар (x, y).

При использовании больших p подсчёт обобщенного символа Якоби в общем случае - очень трудная задача. В некоторых случаях её решение даёт решение задачи дискретного логарифмирования ([123] стр. 182). Однако для верхних и нижних индексов специального вида она может оказаться простой.

Рассмотрим целые алгебраические числа в простом круговом поле $\mathbb{Q}(\xi), \xi = e^{\frac{2\pi i}{p}}$, p - простое. Пусть $\lambda = 1 - \xi$. Рассмотрим мультипликативно независимые в $\mathbb{Z}[\xi]$ элементы $\eta_i = 1 - \lambda^i, i \geq 1$. Известно, что эти элементы, вместе с некоторой степенью первообразного корня по модулю p образуют мультипликативный базис кольца $\mathbb{Z}[\xi]$, и для любого $z \in \mathbb{Z}$ имеем

$$1 - z\lambda \equiv {\eta_1}^{e_1(z)} \cdot \dots \cdot {\eta_{p-1}}^{e_{p-1}(z)} (\mathrm{mod} \lambda^p), \tag{3.18}$$

где $0 \leq e_i \leq p - 1$.

Гипотеза.(опубликована в [38]) Пусть $z \in \mathbb{Z}$. $e_n(z)$ является многочленом от z, в который z входит только в степенях, являющихся делителями n. В случае простого $n = q$ верна формула:

$$e_q(z) \equiv \frac{1}{q}(z^q - z)(\mathrm{mod} p).$$

Эта гипотеза недавно была доказана Раинчик В.С. [39].

В качестве итога данной главы можно назвать то, что найдена операция для схемы типа Сидельникова, стойкая относительно элементарных атак, а также достаточно легко вычислимая при некоторых значениях аргумента вида (3.18).

Помимо открытого распределения ключей новые конструкции можно предложить и для использования в протоколах шифрования и некоторых других. Этому посвящены статьи [41, 42].

Глава 4

Задача Бризолиса

В первой главе диссертации анализировалось использование задачи дискретного логарифмирования в схемах открытого распределения ключей. В этой главе будет проанализировано использование этой задачи в схемах шифрования и электронной подписи Эль-Гамаля. Задача вскрытия этих схем сведена к решению некоторого сравнения. Для этого сравнения получены некоторые оценки на число решений. По мнению автора, изложенный в данной главе подход следует учитывать при анализе стойкости криптографических протоколов, построенных на основе схемы Эль-Гамаля. В частности, используя результаты этой главы, можно выбором параметров снижать число решений указанного сравнения.

В 1985 году Эль-Гамаль [8] предложил оригинальные схемы шифрования и электронной подписи. Самую большую известность приобрела схема электронной подписи, на основе которой в дальнейшем было создано очень много криптографических протоколов для решения различных конкретных задач защиты информации. Один из примеров - стандарт электронной подписи ЦБ РФ, описанный ниже.

Пусть p – простое число, $GF(p)^* = \langle g \rangle$, то есть g – первообразный корень по модулю p.

A **B**

$k \in_R \{1, ..., p-1\}$,
$y \equiv g^k \pmod{p}$

$- - - - - y - - - - - >$

$r \in_R \{1, ..., p-1\}$,
$t \equiv g^r \pmod{p}$
пусть m – сообщение,
s – решение сравнения
$rs + kt \equiv m \pmod{p-1}$,
$s \in \{1, ..., p-1\}$.

$\xrightarrow{m,(s,t)}$

$t \overset{?}{\in} \{1, ..., p-1\}$,
$s \overset{?}{\in} \{1, ..., p-1\}$,
$g^m \overset{?}{\equiv} y^t t^s \pmod{p}$.

Если последние проверки выполнены, то подпись принимается, в противном случае не принимается.

Подпись сообщения m, то есть пару (s, t), может проверить любой желающий, так как все необходимые для этого параметры передаются по открытому каналу связи и находятся в открытом доступе.

Конечно, умение подписывать в этом протоколе не сложнее задачи дискретного логарифмирования. Однако вопрос об эквивалентности этих задач пока открыт.

DSA (стандарт электронной подписи США)

Пусть [**?**] p, q – простые числа, $q|p-1$, $g \in (\mathbb{Z}/p\mathbb{Z})^*$ – элемент порядка q, сообщение m лежит в некотором множестве сообщений M. Хеш-функция h :

$M \to \mathbb{Z}_q^*$.

A		**B**
$x \in_R \mathbb{Z}/q\mathbb{Z}$,		
$y \equiv g^x \pmod p$,		
	$\xrightarrow{\quad y \quad}$	
$u \in_R \mathbb{Z}/q\mathbb{Z}\backslash\{1\}$,		
$r \equiv g^u \pmod p \pmod q$,		
$s \equiv u^{-1}(h(m) + xr) \pmod q$,		
Проверка $(r, s) \neq (0, 0)$.		
	$\xrightarrow{\quad (m,s,r) \quad}$	
		$v \equiv s^{-1} \pmod q$,
		$0 < r < q$?
		$0 < s < q$?
		$r \stackrel{?}{\equiv} g^{vh(m)} y^{vr} \pmod p \pmod q$.

Здесь под первым взятием по модулю подразумевается получение наименьшего положительного вычета.

Для построения цифровой подписи в ГОСТе Центрального банка РФ 1996 года используется аналогичная схема, где выражение для получения s заменено на следующее: $s \equiv uh(m) + xr \pmod q$, а соответствующее проверочное равенство выглядит как $r \stackrel{?}{\equiv} g^{sh(m)^{-1}} y^{-rh(m)^{-1}} \pmod p \pmod q$.

Отметим [19, 20] важность проверки неравенств в последнем шаге работы ГОСТа. Дело в том, что, решив с помощью китайской теоремы об остатках при некотором c следующую систему

$$\begin{cases} r \equiv c \pmod{q} \\ r \equiv R^c \pmod{p}, \end{cases}$$

где $R \equiv (gy^{-1})^{h(m)^{-1}}$, мы получим пару $(s \equiv r \pmod{q}, r)$, удовлетворяющую проверочному соотношению, причём $0 < r < pq, 0 < s < q$.

Заметим также, что выполнение указанной системы при некотором c эквивалентно выполнению сравнения

$$r \equiv R^r \pmod{p}. \tag{4.1}$$

Значит, умение решать это сравнение относительно $r \in \{1, \ldots, q-1\}$ даст возможность подписывать любое сообщение, то есть осуществлять универсальную подделку. Отметим, что решение рассматриваемого сравнения не сводится сразу к задаче дискретного логарифмирования, хотя, с другой стороны, с помощью Китайской теоремы об остатках оно может быть легко решено для $r \in \{1, \ldots, pq-1\}$.

Аналогичные рассмотрения для схем цифровой подписи Эль-Гамаля и DSA приводят к сравнению вида $r \equiv CR^r \pmod{p}, r \in \{1, \ldots, q-1\}$, что снова приводит к сравнению (4.1) с условием $r = C_1 r', r' \in \{1, \ldots, q-1\}$.

В книге Р.К. Ги "Нерешённые задачи теории чисел" [46] приводится следующая задача, называемая задачей Бризолиса: для каких простых p существует первообразный корень $g(modp)$ и целое $x : 1 \leq x \leq p-1$ такое, что

$$x \equiv g^x \pmod{p}?$$

В 2003 году появилась диссертация М.Л.Кэмпбелл "О неподвижных точках дискретных логарифмов" [110], которая посвящена доказателству того, что для любого простого p существует первообразный корень g по модулю p, взаимнопростой с $p-1$, то есть

$$(g, p-1) = 1. \tag{4.2}$$

Эти первообразные корни дают решение сравнения

$$x \equiv R^x \pmod{p}; x, R \in \mathbb{Z}, (r, p) = 1, \tag{4.3}$$

в виде $x = g, R = g^{g^{-1} \pmod{p-1}}$, где R, очевидно, также первообразный корень, то есть решение задачи Бризолиса. Доказательство использует оценки тригонометрических сумм, в частности оценку Полиа-Виноградова (см.[126], глава 23), а также объёмные компьютерные вычисления. Вопрос о построении конкретных решений, а также оценке их количества, тем самым, остался открытым.

В данной главе [45] получены некоторые общие решения уравнения (4.3), а также получена оценка снизу на число первообразных корней, удовлетворяющих условию (4.2) для случая, когда $p-1$ не содержит "маленьких" нечётных простых множителей.

Будем обозначать $a \pmod{p}$ наименьший положительный вычет по модулю p.

Лемма 4.1. *[45] Пусть* $d \mid p-1$

1) Уравнение (4.3) не имеет решений среди пар (x, R), *таких, что* $ord_p R \mid d$, *тогда и только тогда, когда для любого* $u_0 \in \{1, \ldots, d-1\}$ *выполнено*

$$(d, g^{u_0 \frac{p-1}{d}} \pmod{p}) \nmid u_0.$$

2)Уравнение (4.3) не имеет решений среди пар (x, R), *таких, что* $ord_p R = d$, *тогда и только тогда, когда для любого* $u_0 \in \{1, \ldots, d-1\}$ *выполнено* $(d, g^{u_0 \frac{p-1}{d}} \pmod{p}) \neq (u_0, d)$.

Доказательство. Если записать

$$\begin{cases} x \equiv g^u \pmod p, u \in \{1, \ldots, p-1\} \\ R \equiv g^{\frac{p-1}{d}k} \pmod p, k \in \{1, \ldots, d-1\}, \end{cases}$$

то сравнение (4.1) равносильно тому, что $u \equiv \frac{p-1}{d}k(g^u \pmod p) \pmod{p-1}$.

Отсюда следует, что $u = u_0\frac{p-1}{d}, u_0 \in \{1, \ldots, d-1\}$ и

$$u_0 \equiv k(g^{u_0\frac{p-1}{d}} \pmod p) \pmod d. \tag{4.4}$$

Значит отсутствие решений уравнения (4.3) относительно (x, R) означает отсутствие решений сравнения (4.4) относительно $k, u_0 \in \{1, \ldots, , d-1\}$ (во втором случае $(k, d) = 1$).

Теорема 4.1. *[45] Пусть p, d - простые, $d \mid p-1, d \geq \sqrt{p}+1$. Тогда существует (x, R) - решение уравнения (4.3), такое, что $ord_p R = d$.*

Доказательство. Согласно лемме 4.1, в случае противного имеем

$$(d, g^{u_0\frac{p-1}{d}} \pmod p) \neq (u_0, d)$$

для любого $u_0 \in \{1, \ldots, d-1\}$, то есть, ввиду простоты d,

$$d \mid g^{u_0\frac{p-1}{d}} \pmod p$$

для любого $u_0 \in \{1, \ldots, d-1\}$. Поэтому $d(d-1) \leq p-1$, откуда $d^2 - 2d + 1 < p$, или $d - 1 < \sqrt{p}$, что противоречит условию.

Замечание. *При выполнении условий теоремы 4.1 вероятность найти u_0, такое, что $d \nmid g^{u_0\frac{p-1}{d}} \pmod p$, случайным перебором $u_0 \in \{1, \ldots, d-1\}$ не меньше*

$$\frac{d - 1 - \frac{p-1}{d}}{d-1} = 1 - \frac{p-1}{d(d-1)},$$

что, например, для чисел Софи-Жермен ($p = 2d-1$, d - простое) очень близко к единице.

Далее по u_0 будем искать u, x, R.

Теорема 4.2. *[45] Пусть $p = 2^s q + 1 > 2^{2^s}$, где p - простое, и выполнено одно из двух условий:*

1) q - простое нечётное, или

2) q - любое нечётное, $q \mid ord_p 2$.

Тогда сравнение (4.3) выполнено при

$$x = 2^{2^s}, R \equiv g^{qt+ind_g 2(\frac{q+1}{2})^{2^s-s}} \pmod p$$

для любого g - первообразного корня по модулю p, и $t \in \mathbb{N}$. Кроме того, для t, при которых показатель в последнем сравнении нечётный, R будет первообразным корнем по модулю p.

Доказательство. В условиях теоремы выполнено $q \nmid ind_g 2$, так как иначе $2^{2^s} \equiv g^{2^s ind_g 2} \equiv 1 \pmod p$, что противоречит условию. Поэтому в случае 1) $(q, ind_g 2) = 1$. В случае 2) это же следует из того, что при $\frac{p-1}{2^s} \mid ord_p 2 = \frac{p-1}{(ind_g 2, p-1)}$ выполнено $(ind_g 2, p-1) = 2^t, t \in \{1, \ldots, s\}$.

Пусть $u_0 \equiv 2^s ind_g 2 \pmod{p-1}$ или $2^{2^s} \equiv g^{u_0} \pmod p$, откуда соответственно $(g^{u_0} \pmod p, p-1) = 2^s, (u_0, p-1) = 2^s$. Поэтому существует $k, (k, p-1) = 1$, удовлетворяющее сравнению $u_0 \equiv k(g^{u_0} \pmod p) \pmod{p-1}$, которое равносильно сравнению $g^{u_0} \equiv g^{k(g^{u_0} \pmod p)} \pmod p$, или

$$k \equiv \frac{u_0}{2^s}(2^{2^s-s})^{-1} \pmod{\frac{p-1}{2^s}},$$
$$k \equiv \frac{u_0}{2^s}(\frac{q+1}{2})^{2^s-s} \pmod q,$$
$$k \equiv ind_g 2(\frac{q+1}{2})^{2^s-s} \pmod q,$$

откуда

$$R \equiv g^k \equiv g^{ind_g 2(\frac{q+1}{2})^{2^s-s}+qt} \pmod p$$

для любого целого t удовлетворяет сравнению (4.3) при $x = 2^{2^s}$.

Легко видеть, что если 2 является первообразным корнем по модулю p, то $s \in \{1,2\}$, ($s=1$ выполнено $q \equiv 1 \pmod 4$), иначе 2 является квадратичным вычетом по модулю p. Поэтому верно следующее замечание:

Замечание. *Условию теоремы 4.2 удовлетворяют, например, простые $p>5$, для которых 2 является первообразным корнем.*

Действительно, из вышеизложенного следует, что в этих условиях при $p>16$ выполнено условие теоремы 4.2. Легко видеть, что для $p \in \{13,11,7\}$ число 2 является первообразным корнем только при $p=11$, и в этом случае условие теоремы 4.2 также выполняется.

Теорема 4.3. *[45] Пусть $p \geq 5$ простое, тогда алгоритм случайного перебора нечётных чисел $g \in \{1,\ldots,p-1\}$ выдаст первообразный корень по модулю p, удовлетворяющий условию (4.2) с вероятностью не меньше*

1) $\frac{3\varphi(p-1)}{p-1} - 1$ *при* $p \equiv 1 \pmod 4$.

2) $\frac{4\varphi(p-1)}{p-1} - \frac{7}{4} - \frac{1}{2(p-1)}$ *при* $p \equiv -1 \pmod 4$.

Замечание. *Соответствующая оценка для числа таких g есть* $(\frac{3\varphi(p-1)}{p-1}-1)\frac{p-1}{2}$ *и* $(\frac{4\varphi(p-1)}{p-1} - \frac{7}{4} - \frac{1}{2(p-1)})\frac{p-1}{2}$ *соответственно в первом и втором случаях.*

Заметим сразу, что в общем случае соответствующие оценки могут оказаться отрицательными и ничего не дадут, так будет, например, при $p=7$. Однако для большинства чисел, использующихся в криптографии, то есть тех, у которых все простые делители числа $\frac{p-1}{2}$ велики, эти оценки оказываются эффективными, например, для чисел Софи-Жермен ($\varphi(p-1) = \frac{p-1}{2} - 1 = \frac{p-3}{2}$), и элементарные вычисления дают для первого случая теоремы в качестве нижней оценки на вероятность величину $\frac{1}{2} - \frac{3}{p-1}$, а в качестве нижней оценки на количество рассматриваемых первообразных корней — величину $\frac{p-7}{4}$. Для второго случая получаем соответственно $\frac{1}{4} - \frac{9}{2(p-1)}$ и $\frac{p-19}{8}$. Заметим, что оценки на вероятность при этом всегда положительные, за исключением случаев $p \in \{5,7,11\}$.

Поскольку искомые первообразные корни для этих значений p существуют (равны соответственно $3, 5$ и 7), то из рассматриваемой теоремы следует:

Следствие 4.1. *[45] Гипотеза Бризолиса верна для всех простых Софи-Жермен.*

Легко видеть, что аналогичное верно и для простых Ферма, больших 3, так как в этом случае $\varphi(p-1) = \frac{p-1}{2}$.

Заметим также, что аналогичные оценки работы [47] асимптотически лучше полученных в этой теореме для случая $p \equiv -1 \pmod 4$, правда, эффективная граница, которую удаётся вычислить, используя параметры работы [47], оказывается довольно большой (несколько тысяч десятичных знаков).

Доказательство теоремы 4.3. 1). В рассматриваемом случае следующие равенства равносильны

$$(k, p-1) = 1, (k + \frac{p-1}{2}, p-1) = 1.$$

Значит число g является первообразным корнем по модулю p тогда и только тогда, когда им является число $p - g$. Значит, чётных и нечётных первообразных корней поровну, то есть $\frac{\varphi(p-1)}{2}$. Пусть $M = \{x \mid x \in \{1, \ldots, p-1\}, x$ -нечётное, $(x, p-1) > 1\}$. Тогда $\sharp M = \frac{p-1}{2} - \varphi(p-1)$. Поэтому вероятность того, что g — первообразный корень и $(g, p-1) = 1$, не меньше

$$\frac{\frac{\varphi(p-1)}{2} - [\frac{p-1}{2} - \varphi(p-1)]}{\frac{p-1}{2}} = \frac{3\varphi(p-1)}{p-1} - 1.$$

2). В этом случае $p = 2q + 1$, где q - нечётное. В частности, (-1) - невычет по модулю p.

Назовём точным вычетом по модулю p число x, для которого $(ind^p x, p-1) = 2$.

В рассматриваемом случае следующие равенства равносильны

$$(k, p-1) = 1, (k + \frac{p-1}{2}, p-1) = 2.$$

Значит, число g является первообразным корнем по модулю p тогда и только тогда, когда $p-g$ - точный вычет. Поэтому $\varphi(p-1)$ - число точных, а $\frac{p-1}{2}-\varphi(p-1)$ - число неточных вычетов. Обозначим S число нечётных первообразных корней (или чётных точных вычетов). Рассмотрим два случая.

Пусть сначала 2 - невычет (то есть $\frac{p-1}{2} \equiv 1 \pmod 4$). Если $g = 2k(g) \in \{1, \ldots, p-1\}$ - чётный первообразный корень (число таких g равно $\varphi(p-1)-S$), то $k(g) \in \{1, \ldots, \frac{p-1}{2}\}$ - вычет. Отсюда, кстати, сразу следует, что $\varphi(p-1) - S \leq \frac{p-1}{2}$, то есть $S \geq \varphi(p-1) - \frac{p-1}{2}$, однако эта оценка тривиальна, так как правая её часть отрицательна. Среди $\varphi(p-1)-S$ таких $k(g)$ точных вычетов как минимум $\varphi(p-1) - S - (\frac{p-1}{2} - \varphi(p-1)) = 2\varphi(p-1) - \frac{p-1}{2} - S$. Из них как минимум $2\varphi(p-1) - \frac{p-1}{2} - 2S$ нечётных. Поэтому число $\frac{\frac{p-1}{2}-1}{2} + 1 = \frac{p+1}{4}$ нечётных чисел среди $\{1, \ldots, \frac{p-1}{2}\}$ не меньше $2\varphi(p-1) - \frac{p-1}{2} - 2S$. Поэтому

$$2S \geq 2\varphi(p-1) - \frac{3p-1}{4},$$

то есть

$$S \geq \varphi(p-1) - \frac{3p-1}{8}.$$

Значит, вероятность найти первообразный корень, удовлетворяющий условию (4.2) (то есть $g \notin M$), перебирая нечётные числа от 1 до $p-1$, не меньше

$$\frac{S - [\frac{(p-1)}{2} - \varphi(p-1)]}{\frac{p-1}{2}} \geq \frac{2\varphi(p-1) - \frac{7p-5}{8}}{\frac{(p-1)}{2}} = \frac{4\varphi(p-1)}{p-1} - \frac{7}{4} - \frac{1}{2(p-1)}.$$

Пусть теперь 2 - вычет (то есть $\frac{p-1}{2} \equiv (-1) \pmod 4$). В этом случае соответствующие $k(g)$ - невычеты, а $p - k(g) \in \{\frac{p+1}{2}, \ldots, p-1\}$ - вычеты. Далее рассуждения проводятся аналогично с заменой $k(g)$ на $p - k(g)$.

Глава 5

Гиперэллиптические кривые

5.1 Арифметика дивизоров

Пусть $\mathbb{F}[x], \mathbb{F}(x)$ соответственно кольцо многочленов и поле рациональных функций над полем $\mathbb{F}$.

Элемент называется алгебраическим, если он алгебраический над полем рациональных чисел $\mathbb{Q}$. В этом случае, очевидно, он является корнем многочлена с целыми коэффициентами.

Элемент α называется целым над кольцом $\mathbb{K}$, если он является корнем многочлена с коэффициентами из $\mathbb{K}$ со старшим коэффициентом равным единице.

Если в этом определении $\mathbb{K} = \mathbb{Z}$, то говорят о целых алгебраических числах. При помощи теоремы о симметрических многочленах нетрудно показать, что сумма и произведение двух алгебраических элементов, а также обратный элемент к любому алгебраическому также является алгебраическим. Таким образом, алгебраические над фиксированным полем элементы также образуют поле. Аналогично доказывается, что целые элементы над фиксированным кольцом образуют кольцо. Подробные доказательства можно найти, например, в [3].

Кольцо $\mathbb{K}$ называется целозамкнутым в некотором объемлющем кольце $\mathbb{L}$, если каждый элемент $\alpha \in L$, целый над $\mathbb{K}$, снова лежит в $\mathbb{K}$.

Для конечных полей и полей нулевой характеристики верна следующая теорема:

Теорема 5.1. *[129](О примитивном элементе) Пусть* $\mathbb{F}(\alpha_1, \ldots, \alpha_h)$ *- конечное алгебраическое расширение поля* $\mathbb{F}$*. Тогда для некоторого алгебраического над* $\mathbb{F}$ *элемента* α

$$\mathbb{F}(\alpha_1, \ldots, \alpha_h) = \mathbb{F}(\alpha).$$

Необходимые для понимания дальнейшего текста свойства дивизоров гиперэллиптических кривых изложены в приложении.

5.1.1 Алгоритмы сложения и приведения дивизоров

Пусть $F_q(x, y)$ - квадратичное расширение поля $F_q(x)$, образованное присоединением алгебраического элемента второй степени y, удовлетворяющего следующему уравнению гиперэллиптической кривой C [127, 128]:

$$y^2 + f(x)y + h(x) = 0, degf(x) \leq g, degh(x) = 2g + 1. \tag{5.1}$$

Пусть $\overline{F}_q$ - алгебраическое замыкание поля F_q, а кривая C является гладкой. Для каждого (дробного) идеала α кольца $\overline{F}_q[x, y]$ определим

$$div\alpha = \Sigma n_P P, \tag{5.2}$$

где точки P соответствуют простым идеалам кольца $\overline{F}_q[x, y]$, которые мы будем обозначать той же буквой, а их кратности n_P определены разложением идеала α в произведение простых идеалов:

$$\alpha = \Pi P^{n_P}. \tag{5.3}$$

Это определение корректно, так как согласно [129, Теорема 19 гл.V, §8,], кольцо $\overline{F}_q[x, y]$ является областью с однозначным разложением на простые идеалы.

При этом отображение (5.2) задаёт изоморфизм якобиана кривой (5.1), то есть группы дивизоров, и группы (дробных) идеалов кольца $\overline{F}_q[x,y]$, определённый равенством

$$\Phi_{\overline{F}_q(x,y)}(\alpha) = \Sigma n_P P - P_\infty \Sigma n_P, \tag{5.4}$$

где P_∞ - бесконечно удалённая точка (продолжение нормирования, порождённого простым делителем идеала $\frac{1}{x}$), которая в данном случае в силу [48, лемма 2.4 §1 глава 2, часть 1] единственна. Два идеала $\alpha, \beta \subset \overline{F}_q(x,y)$ называются подобными, если существует такой элемент $\gamma \in \overline{F}_q[x,y]$, что $\gamma\alpha = \beta$. Соответствующие им дивизоры (5.2) в рассматриваемом случае будут подобны (их разность является дивизором элемента γ). При этом отображение (5.2) задаёт изоморфизм группы классов подобных дивизоров (якобиан поля $\overline{F}_q(x,y)$) и группы классов подобных идеалов кольца $\overline{F}_q[x,y]$. Подобие идеалов и дивизоров будем, как и раньше, обозначать $\sim$. Все конечные нормирования поля $\overline{F}_q(x)$ соответствуют идеалам вида $(x - x_0)$, где $x_0 \in \overline{F}_q$ (это все простые идеалы кольца $\overline{F}_q[x]$). При расширении кольца $\overline{F}_q[x]$ до кольца $\overline{F}_q[x,y]$, где y удовлетворяет уравнению (5.1), идеал $(x - x_0)$ перестаёт быть простым и раскладывается в произведение двух, возможно, одинаковых простых идеалов:

$$(x - x_0, y - y_0), (x - x_0, y - \overline{y}_0), \tag{5.5}$$

где $\overline{y}_0$ - сопряжённое к y_0 решение уравнения (5.1) при $x = x_0$. Действительно, каждый из них, очевидно, делит идеал $(x - x_0)$, а перемножив эти идеалы при помощи уравнения (5.1), легко увидеть, что их произведение делится на $(x - x_0)$. Кроме того, идеалы (5.5) — простые, так как они состоят из всех многочленов, обращающихся в ноль в точках соответственно $P = P(x_0, y_0)$ и $\overline{P} = P(x_0, \overline{y}_0)$ (в скобках указаны координаты точек). Действительно, для $a(x), b(x) \in \overline{F}_q[x]$ выполнена следующая цепочка следствий:

$$a(x)+yb(x)|_{x=x_0,y=y_0}=0\Rightarrow a(x)+yb(x)-(y-y_0)b(x)|_{x=x_0,y=y_0}=0\Rightarrow$$

$$\Rightarrow a(x)+yb(x)-(y-y_0)b(x)=a(x)+y_0b(x)=(x-x_0)c(x)$$

для некоторого $c(x)\in\overline{F}_q[x]$. Значит, $a(x)+yb(x)\in(x-x_0,y-y_0)$. Поэтому

$$\Phi_{\overline{F}_q(x)}((x-x_0))=Q_{x_0}-P_\infty,$$

где Q_{x_0} - нормирование поля $\overline{F}_q(x)$, отвечающее простому идеалу $(x-x_0)$, а

$$\Phi_{\overline{F}_q(x,y)}((x-x_0))=P_{x_0,y_0}+P_{x_0,\overline{y}_0}-2P_\infty, \tag{5.6}$$

где $P(x_0,y_0),\overline{P}=P(x_0,\overline{y}_0)$ - нормирования поля, отвечающие простым идеалам (5.5). Действительно, согласно [140, Теорема 4, гл. 1] дивизор элемента поля имеет нулевую степень. Коэффициент 2 при бесконечно удалённой точке означает, что идеал $\frac{1}{x}$ в соответствующем кольце нормирования поля $\overline{F}_q(x,y)$ является квадратом простого идеала.

В дальнейшем мы будем говорить только о дивизорах нулевой степени и бесконечно удалённую точку будем опускать.

Дивизор $D=\sum m_PP$ поля $\overline{F}_q(x,y)$ определён над F_q, если $D^\sigma=\Sigma m_PP^\sigma=D$ для любого $\sigma\in Gal(\overline{F}_q/F_q)$ или $D=\sum_k m_k\sum_\sigma {P_\sigma}^k$.

Приведём некоторые примеры. При $x_0\in F_q$ дивизор (5.6) определён над F_q. Определёнными над F_q будут дивизоры $\sum_\sigma\Phi_{\overline{F}_q(x)}(x-{x_0}^\sigma),\sum_\sigma\Phi_{\overline{F}_q(x,y)}(x-{x_0}^\sigma)$. Отсюда, очевидно, следует:

Замечание. *Любой, определённый над F_q, дивизор поля эквивалентен некоторому полуприведённому дивизору $D=\sum m_PP$, определённому над F_q, то есть для всех конечных точек P выполнено: $m_P\geq 0$, а если $m_P>0$, то $m_{\overline{P}}=0$ при $\overline{P}\neq P, m_{\overline{P}}=1$, при $\overline{P}=P$ (см. также [60, Appendix Л. 4.2]).*

Пусть даны два дивизора: $D_1=\sum m_PP, D_2=n_PP$. Их наибольшим общим делителем называется дивизор $(D_1,D_2)=\sum min(m_P,n_P)P$.

Теорема 5.2. *([60, Appendix Теорема 5.1]) Пусть* $D = \sum m_i P_i$ - *полуприведённый дивизор,* $P_i = P_i(x_i, y_i); x_i, y_i \in \overline{F}_q; a(x) = \prod (x - x_i)^{m_i}$. *Тогда существует единственный многочлен* $b(x) \in \overline{F}_q[x]$ *такой, что выполнены следующие условия:*

1. $degb(x) < dega(x)$,
2. $b(x_i) = y_i$ *для всех* i , *для которых* $m_i \neq 0$,
3. $a(x)|b(x)^2 + b(x)f(x) + h(x)$.

При этом $D = (\Phi_{\overline{F}_q(x,y)}(a(x)), \Phi_{\overline{F}_q(x,y)}(y - b(x)))$. *В дальнейшем будем обозначать*

$$D = div(a, b). \tag{5.7}$$

Некоторые из представленных в этом разделе теорем доказаны в [60], соответствующие ссылки указаны. Однако эти доказательства используют явное представление дивизоров в виде формальных сумм и достаточно громоздки. Приведённые ниже доказательства [48] используют только что введённое полиномиальное представление и занимают гораздо меньше места.

Лемма 5.1. *Кратность вхождения корней многочлена* f *в многочлен* $B^2 + fB + h$ *не превосходит единицу.*

Доказательство. Действительно, в противном случае следующая система разрешима:

$$\begin{cases} F(x, B(x)) = B^2 + fB + h = 0 \\ F'_x(x, B(x)) = f'B + h' = 0 \\ F'_y(x, B(x)) = f = 0, \end{cases}$$

что противоречит гладкости кривой $F(x, y) = y^2 + f(x)y + h(x) = 0$. Лемма 5.1 доказана.

Теорема 5.3. *Пусть* $a(x), b(x) \in \overline{F}_q[x]$. *Тогда* $div(a,b) = \Phi_{\overline{F}_q(x,y)}(a(x), y - b(x))$ *- полуприведённый дивизор.*

Доказательство. Пусть имеется два разложения в произведение простых идеалов

$$\alpha = \prod P_i^{a_i}, \beta = \prod P_i^{b_i},$$

тогда сумма этих идеалов, другими словами, их наибольший общий делитель равен

$$(\alpha, \beta) = (\prod P_i^{a_i}, \prod P_i^{b_i}) = \prod P_i^{min(a_i,b_i)} (\prod P_i^{a_i - min(a_i,b_i)}, \prod P_i^{b_i - min(a_i,b_i)}).$$

Последний сомножитель является единицей, то есть совпадает со всем кольцом, в противном случае он делится на некоторый простой идеал, другими словами, содержится в нём. Значит, в этом идеале содержатся и стоящие в скобках произведения, то есть они оба делятся на него, что невозможно. Поэтому

$$\Phi_{\overline{F}_q(x,y)}((\alpha, \beta)) = \sum min(a_i, b_i) \Phi_{\overline{F}_q(x,y)}(P_i) =$$

$$= \text{НОД } (\Phi_{\overline{F}_q(x,y)}(\alpha), \Phi_{\overline{F}_q(x,y)}(\beta)).$$

В частности, $\Phi_{\overline{F}_q(x,y)}(a(x), y - b(x)) = div(a,b)$. Если $P = P(x_0, y_0) \neq \overline{P}$, то она не может принадлежать спектру этого дивизора одновременно с $\overline{P}$. Если же $P = \overline{P}$ лежит в спектре, то $f(x_0) = 0$, и, по лемме 5.1, $ord_P(y - b) = deg_{x - x_0}(b^2 + fb + h) = 1$.

Теорема 5.3 доказана. Из определений ясно, что дивизор вида (5.7) определён над F_q тогда и только тогда, когда $a(x), b(x) \in F_q[x]$.

Теорема 5.4.

$$-div(a,b) \sim div(a, -f - b).$$

Доказательство. Если $b(x_i) = y_i$, то $-f(x_i) - b(x_i) = \overline{y}_i$. Поэтому если в спектре $div(a,b)$ лежит некоторая точка, то сопряжённая к ней лежит в спектре $div(a,-f-b)$, а кратность будет одинаковой, поскольку она определяется многочленом $a(x)$. Теорема 5.4 доказана.

Алгоритм сложения дивизоров

Вход: полуприведённые дивизоры вида (5.7), определённые над F_q : $D_1 = div(a,b), D_2 = div(c,d)$.

Выход: полуприведённый дивизор $D = div(s,t)$, определённый над F_q : $D \sim D_1 + D_2$.

1 шаг: При помощи алгоритма Евклида получим $e_1, e_2, l_1 \in F_q[x]$, такие, что

$$l_1 = (a,c) = e_1 a + e_2 c.$$

2 шаг: При помощи алгоритма Евклида получим $g_1, g_2, l \in F_q[x]$, такие, что

$$l = (l_1, b+d+f) = g_1 l_1 + g_2(b+d+f).$$

3 шаг: Пусть $v_1 = g_1 e_1, v_2 = g_1 e_2, w = g_2$, тогда $l = v_1 a + v_2 c + w(b+d+f)$.

4 шаг: Вычислить $s = ac/l^2, t = (v_1 ad + v_2 cb + w(bd-h))/l(mods)$.

Обоснование алгоритма сложения дивизоров

Теорема 5.5. *D - полуприведён, $D \sim D_1 + D_2$.*

Доказательство.

Рассмотрим произведение идеалов:

$$\begin{aligned}(a, y-b)(c, y-d) &= (ac, ay-ad, cy-cb, y^2 - y(b+d) + bd) = \\ &= (ac, ay-ad, cy-cb, -y(b+d+f) + bd - h).\end{aligned} \tag{5.8}$$

Рассмотрим построенные в алгоритме элементы $v_1, v_2, w, l \in F_q[x]$:

$$l = (a, c, b + d + f) = v_1 a + v_2 c + w(b + d + f).$$

Определим

$$\begin{aligned}
&\psi_0 = ac + v_1(ay - ad) + v_2(cy - cb) - w(-y(b + d + f) + bd - h) = \\
&= ac - v_1 ad - v_2 cb + w(bd - h) + ly = ly - t', \\
&t' = v_1 ad + v_2 cb + w(bd - h) - ac, \\
&\psi_1 = ay - ad - \psi_0 a/l = -ad + t'a/l \in F_q[x], \\
&\psi_2 = cy - cb - \psi_0 c/l = -cb + t'c/l \in F_q[x], \\
&\psi_3 = -y(b + d + f) + bd - h + \psi_0(b + d + f)/l = bd - h - t'(b + d + f)/l \in F_q[x], \\
&s' = (\psi_1, \psi_2, \psi_3)
\end{aligned} \tag{5.9}$$

Идеал (5.8), очевидно, равен $(\psi_0, \psi_1, \psi_2, \psi_3) = (\psi_0, s') = (ly - t', s')$. Обозначим N норму дробного идеала в $F_q(x, y)$ над $F_q(x)$. При этом под нормой элемента будем понимать норму соответствующего главного идеала. При этом

$$N(a(x) + yb(x)) = (a(x) + yb(x))(a(x) + \overline{y}b(x)) = a(x)^2 + a(x)b(x)f(x) + b(x)^2 h(x).$$

Тогда

$$ac | (b^2 + bf + h)(d^2 + df + h) = N((y - b)(y - d)) =$$

$$= N(-(b + d + f)y + bd - h) = (bd - h)^2 - (bd - h)(b + d + f)f + (b + d + f)^2 h.$$

Отсюда видно, что $l = (a, c, b + d + f) | bd - h$. Значит, по построению получим $l | t'$, поэтому из (5.9) $l | \psi_i$ для $i = 1, 2, 3$, откуда $l | s'$. Поэтому идеал (5.8) равен

$$(a, y - b)(c, y - d) = l(s'', y - t''), \tag{5.10}$$

где $s'' = s'/l, t'' = t'/l$. Пусть $s = (s'', (t'')^2 + ft'' + h)$ - наибольший общий делитель двух многочленов, тогда

$$(s'', y - t'') = (s, y - t''). \tag{5.11}$$

Это равенство выполнено ввиду того, что для любого простого идеала p , для которого степени его вхождения в s и s'' удовлетворяют неравенству $\gamma_p(s) < \gamma_p(s'')$, выполнено $\gamma_p(s) = \gamma_p((t'')^2 + ft'' + h) = \gamma_p((y - t'')(-y - f - t'')) \geq \gamma_p(y - t'')$.

Согласно [130, гл. 1, следствие 3, предложение 21], для $A, B \in F_q[x]$ выполнено $N(A, y - B) = (A, y - B)(A, \overline{y} - B)$, где $\overline{y} = -f - y$.

Лемма 5.2. *В наших обозначениях*

$$N(A, y - B) = (A, N(y - B))$$

(справа стоит наибольший общий делитель многочленов A *и* $N(y - B) = B^2 + fB + h$ *).*

Доказательство. Из Леммы 5.1 получим, что при $A' = (A, B^2 + fB + h)$ следующие три многочлена не имеют общих корней:

$$A', f, \frac{B^2 + fB + h}{A'},$$

то есть их наибольший общий делитель тривиален:

$$(A', f, \frac{B^2 + fB + h}{A'}) = 1.$$

Также как и для равенства (5.11), получим

$$N(A, y - B) = N(A', y - B) = (A', y - B)(A', -y - f - B)$$

$$= (A'^2, A'(y - B), A'(-y - f - B), B^2 + fB + h)$$

$$= (A'^2, A'(y - B), A'(-f), B^2 + fB + h)$$

$$= A'(A', f, \frac{B^2 + fB + h}{A'}, y - B) = A'.$$

Лемма 5.2 доказана.

Теперь мы можем завершить обоснование алгоритма. Переходя в равенстве (5.10) с учётом (5.11) к нормам, получим равенство главных идеалов:

$$(a)(c) = l^2(s),$$

откуда $s = \frac{ac}{l^2}$. Для завершения алгоритма остаётся взять $t \equiv t''(mods), degt(x) < degs(x)$. Окончательно получим:

$$\Phi_{\overline{F}_q(x,y)}((a, y-b)(c, y-d)) \sim \Phi_{\overline{F}_q(x,y)}(l(s, y-t'')) \sim \Phi_{\overline{F}_q(x,y)}(s, y-t'') \sim div(s,t)$$

Теорема 5.5 доказана.

Полуприведённый дивизор $D = \sum m_P P$ называется приведённым, если $\sum m_P \leq g$.

Алгоритм приведения дивизоров

Вход: полуприведённый дивизор вида (5.7), определённый над F_q : $D = div(a, b)$.

Выход: приведённый дивизор вида (5.7) $D' = div(a', b')$, определённый над $F_q : D \sim D'$.

1 шаг: вычислить $a_1(x) = b(x)^2 + b(x)f(x) + h(x)/a(x), b_1(x) = -b(x) - f(x)(moda_1(x))$,

$degb_1(x) < dega_1(x)$.

2 шаг: если $dega_1(x) > g$, присвоить $a(x) = a_1(x), b(x) = b_1(x)$, идти на шаг 1. Иначе разделить $a_1(x)$ на его старший коэффициент, вывод: $D' = div(a_1, b_1)$.

Теорема 5.6. *D' - приведён, $D \sim D'$.*

Доказательство. Доказательство эквивалентности выразим следующей цепочкой равенств:

$$(-y - f(x) - b(x))(a(x), y - b(x)) =$$
$$= (a(x)(-y - f(x) - b(x)), b(x)^2 + b(x)f(x) + h(x)) =$$
$$= (a(x)(y + f(x) + b(x)), a(x)a_1(x)) = a(x)(a_1(x), y + b(x) + f(x)) =$$
$$= a(x)(a_1(x), y - b_1(x)) = a(x)div(a_1(x), b_1(x)).$$

Согласно теореме 5.3, на каждом шаге работы алгоритма получается пара многочленов, задающая полуприведённый дивизор. Поэтому остаётся доказать, что $dega_1(x) < dega(x)$.

$$dega(x) + dega_1(x) \leq max(2degb(x), degb(x) + g, 2g + 1) \leq$$
$$\leq max(2dega(x) - 2, dega(x) + g - 1, 2g + 1).$$

Если $dega(x) \geq g + 1$, то

$$dega_1(x) \leq max(dega(x) - 2, g - 1, g) = max(dega(x) - 2, g).$$

Поэтому в случае, когда $dega_1(x) > g$, имеем $dega_1(x) \leq dega(x) - 2$.

Теорема 5.6 доказана.

Из этой теоремы и замечания 5.1.1 следует:

Теорема 5.7. *Для любого дивизора нулевой степени существует единственный эквивалентный ему приведённый дивизор. Алгоритм приведения показывает, что это утверждение остаётся верным, если оба дивизора были определены над* F_q .

Теорема 5.8. *[48] Если два приведённых дивизора эквивалентны, то их приведённые формы вида (5.7) совпадают.*

Доказательство.

Пусть имеется два эквивалентных приведённых дивизора вида (5.7):

$$(a, y-b)\ (a', y-b');$$

$$a, a', b, b' \in \overline{F}_q[x], dega' \leq dega \leq g, degb < dega, degb' < dega'.$$

Тогда для некоторых многочленов $c, d, c', d' \in \overline{F}_q[x]$ имеем:

$$(c+dy)(a, y-b) = (c'+d'y)(a', y-b').$$

Умножая на сопряжённое к $(c'+d'y)$, получим для некоторых $c_1, d_1, e_1 \in \overline{F}_q[x]$:

$$(c_1+d_1y)(a, y-b) = e_1(a', y-b'). \tag{5.12}$$

Если многочлен $R(x)$ делит многочлен $A(x)+B(x)y$, то для некоторых $A'(x), B'(x) : A+By = R(A'+B'y)$, откуда, ввиду того, что $y \notin \overline{F}_q(x)$, получим $A = RA', B = RB'$. Поскольку из равенства (5.12) для наибольшего общего делителя (c_1, d_1) многочленов c_1, d_1 имеем $(c_1, d_1)|e_1(y-b')$, откуда $(c_1, d_1)|e_1$, всё равенство (5.12) можно на него сократить. В дальнейшем будем считать, что $(c_1, d_1) = 1$. Аналогично $e_1|a(c_1+d_1y)$, откуда $e_1|a(c_1, d_1) = a$. Взяв норму от обеих частей равенства (5.12), получим

$$N(c_1+d_1y)a = e_1^2a',$$

или

$$(c_1^2 + c_1d_1f + d_1^2h)a = e_1^2a'. \tag{5.13}$$

Пусть $d_1 \neq 0$. Рассмотрим случай $degc_1 > g + degd_1$, тогда

$$degc_1^2 \geq 2g + 2degd_1 + 2 > degd_1^2h, degc_1^2 > degc_1 + g + degd_1 \geq degc_1d_1f,$$

поэтому $deg(c_1^2 + c_1 d_1 f + d_1^2 h) = deg c_1^2 \geq 2g+1$. Аналогично во всех остальных случаях получим $deg(c_1^2 + c_1 d_1 f + d_1^2 h) = deg d_1^2 h \geq 2g+1$. Поэтому сравнивая степени в правой и левой части равенства (5.13), ввиду того, что $dega \geq dega'$, получим $dege_1 \geq g+1$, что противоречит тому, что $e_1|a$ и $dega \leq g$. Поэтому $d_1 = 0$, и из равенства (5.12) получим, как и раньше, $e_1|c_1, c_1|e_1$; откуда $c_1 = e_1$, и всё равенство (5.12) можно на него сократить. Получим

$$(a, y-b) = (a', y-b').$$

Взяв норму от обеих частей, получим $a = a'$. Если теперь умножить обе части этого равенства на идеал $(a, -y-f-b)$, получим, что $a|(y-b')(-y-f-b) = b'(b+f) + y(b'-b-f) + yf + h = (b'b + b'f + h) + y(b'-b)$, откуда, как и раньше, $a|b'-b$, откуда, поскольку $degb, degb' < dega$, получим $b' = b$.

Теорема 5.8 доказана.

Алгоритм разложения дивизора по базе и построение базы дивизоров не являются новыми и изложены в приложении.

5.1.2 Алгоритмы приведения матриц

В некоторых прикладных задачах компьютерной алгебры возникает необходимость приводить матрицы к эрмитовой и смитовой нормальной форме (ЭНФ, СНФ). Например, в методе спуска Вейля на эллиптических кривых над полем характеристики 2, для произвольной квадратной невырожденной матрицы A требуется вычисление не только Смитовой нормальной формы $V = SAT$, но и унимодулярной матрицы S преобразования строк. В этом методе матрица A представляет конечный $\mathbb{Z}$ модуль $\mathbb{Z}^n/L$, так что по столбцам записан базис свободного $\mathbb{Z}$ модуля L ранга n. Для построения Смитовой нормальной формы нужно менять не только базис L, но и базис объемлющего пространства $\mathbb{Z}^n$. За это отвечает унимодулярная матрица S, по строкам

которой записан новый базис $\mathbb{Z}^n$. Сама СНФ задает координаты в этом базисе $\mathbb{Z}$ модуля L. Поэтому вычисление матрицы преобразований столбцов не так актуально. Модулярный алгоритм приведения к эрмитовой нормальной форме хорошо известен [131, 49]. Там же приведён алгоритм приведения к смитовой нормальной форме, однако вычисление матриц преобразований описано только для немодулярных алгоритмов, реализация которых на компьютере практически не возможна из за быстрого роста используемых коэффициентов. В работе [66] построен соответствующий модулярный алгоритм, использующий полиномиальный характер задачи разложения многочленов на множители. В работе [67] предложен асимптотически быстрый модулярный алгоритм приведения целочисленной матрицы к СНФ, позволяющий затем построить матрицы преобразований при помощи нескольких обращений и произведений матриц, накопленных в процессе построения СНФ. В данной статье дано более полное обоснование модулярного алгоритма приведения к эрмитовой нормальной форме чем в [131], позволившее построить новый модулярный алгоритм приведения к смитовой нормальной форме, одновременно вычисляющий и левую матрицу преобразований. Это позволяет сэкономить арифметические операции и необходимую оперативную память. В данной работе не используются дополнительные свойства приводимой матрицы, такие как разреженность, ганкелевость и т.д..

Модулярный алгоритм приведения к ЭНФ

Алгоритм, приведённый в этой главе, взят из [131] для того, чтобы далее привести более полное, чем в [131], его обоснование.

Пусть дана матрица A размера $n \times n, rank A = m, A_i$ - столбцы матрицы A, W_i - столбцы матрицы W, в которую необходимо записать эрмитову нормальную форму матрицы A. D - кратное дискриминанту Z модуля, порождённого столбцами матрицы A. Будем обозначать $u(\mathrm{mod} R)$ наименьший по модулю вычет u по модулю R.

1. $i := m, j := n, k := n, R := D$.

2. Если $j = 1$, идти на шаг 4.

   Иначе, $j := j - 1$, если $a_{i,j} = 0$, идти на шаг 2.

3. При помощи расширенного алгоритма Евклида (РАЕ) вычислить (u, v, d) : $ua_{i,k} + va_{i,j} = d = (a_{i,k}, a_{i,j})$ (в случае $a_{i,k} = d$ должно получаться $u = 1, v = 0$). $B := uA_k + vA_j(\mathrm{mod}R), A_j := (a_{i,k}/d)A_j - (a_{i,j}/d)A_k(\mathrm{mod}R), A_k := B$. Идти на шаг 2.

4. РАЕ $(u, v, d) : ua_{i,k} + vR = d = (a_{i,k}, R), W_i := uA_k(\mathrm{mod}R)$, Если $w_{i,i} = 0$, то $w_{i,i} := R$. Для $j = i + 1, ..., m$ выполнить $q := [w_{i,j}/w_{i,i}], W_j := W_j - qW_i$.

   Если $i = 1$, вывод W и конец, иначе $R := R/d, i := i - 1, k := k - 1, j := k$.

   Если $a_{i,k} = 0$, то $a_{i,k} := R$.

   Идти на шаг 2.

Обоснование алгоритма приведения к ЭНФ

Для простоты рассуждений рассмотрим квадратную матрицу A полного ранга, то есть $m = n$. Именно этот случай используется для построения алгоритма дискретного логарифмирования в нашем случае. Пусть $\mathbf{W} = \text{ЭНФ}\, A$.

В начале работы алгоритма $i = n, R = D$. Пусть на очередном этапе алгоритма $i + 1, ..., n$ строки матрицы A приведены к ЭНФ. В этом случае вычисленное на шаге 4

$$R = D/\mathbf{w}_{i+1,i+1}\cdot, ..., \cdot\mathbf{w}_{n,n}, \tag{5.14}$$

где в знаменателе стоит произведение диагональных элементов матрицы $\mathbf{W}$.

Преобразование столбцов на шаге 3 соответствует умножению справа на целочисленную матрицу с единичным определителем. Редукцию $(\mathrm{mod}\, R)$ здесь можно рассматривать как прибавление столбца, $i + 1, ..., n$ координаты которого равны нулю, а остальные делятся на R.

Пусть $\gamma_i(M)$ - НОД $i \times i$ миноров в последних i строках матрицы M. Обозначим $\tilde{A}$ текущее состояние матрицы A.

Лемма 5.3. *Преобразования, производимые над матрицей $\tilde{A}$ в шаге 3, если их рассматривать без редукции $modR$, не меняют $\gamma_{n-i-1}(\tilde{A})$. Редукция ($modR$), очевидно, не меняет $(\gamma_{n-i-1}(\tilde{A}), R)$.*

Доказательство. Если минор содержал столбцы с номерами k и j, или не содержал ни одного из них, то легко видеть, что его определитель не изменится. Если он содержал столбец с номером k и не содржал столбец с номером j, то преобразования шага 3 сводятся к замене в миноре столбца A_k на столбец $uA_k + vA_j$. Соответствующий определитель $M_k = |..., A_k, ...|_i$ заменится на

$$|..., uA_k + vA_j, ...|_i = u|..., A_k, ...|_i + v|..., A_j, ...|_i.$$

Для миноров, содержащих столбец с номером j и не содержащих столбец с номером k (определитель каждого из них с точностью до знака равен опеделителю минора предыдущего типа, полученного заменой A_k на A_j) определитель $M_j = |..., A_j, ...|_i$ заменится на

$$|..., -\frac{a_{i,j}}{d}A_k + \frac{a_{i,k}}{d}A_j, ...|_i = -\frac{a_{i,j}}{d}|..., A_k, ...|_i + \frac{a_{i,k}}{d}|..., A_j, ...|_i.$$

Итак, M_k, M_j перейдут соответственно в $uM_k \pm vM_j$ и $\pm\frac{a_{i,j}}{d}M_k - \frac{a_{i,k}}{d}M_j$. Их НОД, который будем обозначать $(\cdot, \cdot)$, равен

$$(uM_k \pm vM_j, \mp\frac{a_{i,j}}{d}M_k + \frac{a_{i,k}}{d}M_j) = (M_k, M_j).$$

В этом легко убедится из следующих равенств:

$$\frac{a_{i,k}}{d}(uM_k \pm vM_j) \mp v(\mp\frac{a_{i,j}}{d}M_k + \frac{a_{i,k}}{d}M_j) = (\frac{a_{i,k}}{d}u + \frac{a_{i,j}}{d}v)M_k = M_k,$$

$$\pm\frac{a_{i,j}}{d}(uM_k \pm vM_j) + u(\mp\frac{a_{i,j}}{d}M_k + \frac{a_{i,k}}{d}M_j) = (\frac{a_{i,j}}{d}v + \frac{a_{i,k}}{d}u)M_j = M_j.$$

Вычисляя теперь НОД, группируя предварительно соответствующие миноры разных типов по парам, получим доказательство Леммы 5.3.

В частности после первого перехода на шаг 4

$$(\gamma_1(A), D) = (\gamma_1(\tilde{A}), D) = (a_{n,n}, R). \tag{5.15}$$

Поскольку

$$\gamma_{n-i+1}(\mathbf{W}) = \mathbf{w}_{i,i}\cdot, ..., \cdot\mathbf{w}_{n,n} \mid det\mathbf{W} \mid D,$$

а эрмитовая нормальная форма может быть получена из исходной матрицы при помощи элементарных преобразований столбцов, которые не меняют $\gamma_k(A)$, то при любом i

$$\mathbf{w}_{i,i}\cdot, ..., \cdot\mathbf{w}_{n,n} = \gamma_{n-i+1}(\mathbf{W}) = (\gamma_{n-i+1}(\mathbf{W}), D) = (\gamma_{n-i+1}(A), D). \tag{5.16}$$

При $i = n$ из этого равенства и равенства (5.15) следует, что $w_{n,n} = \mathbf{w}_{n,n}$ вычислено верно.

Пусть теперь $\mathbf{w}_{i+1,i+1}, ..., \mathbf{w}_{n,n}$ уже вычислены, и пусть $a_{i,i}\mathbf{w}_{i+1,i+1}, ..., \mathbf{w}_{n,n}$ — диагональные элементы матрицы $\tilde{A}$, получившиеся после шага 3. Пусть L — верхнетреугольная матрица, вычисленная рассматриваемым алгоритмом с постоянным $R = D$, тогда, опять используя (5.16) и Лемму 5.3, получим

$$\mathbf{w}_{i+1,i+1}\cdot, ..., \cdot\mathbf{w}_{n,n} = (\gamma_{n-i}(A), D) = (\gamma_{n-i}(L), D) = (l_{i+1,i+1}\cdot, ..., \cdot l_{n,n}, D),$$

откуда и из (5.14) $1 = (\frac{l_{i+1,i+1}\cdot,...,\cdot l_{n,n}}{\mathbf{w}_{i+1,i+1}\cdot,...,\cdot\mathbf{w}_{n,n}}, R)$ аналогично

$$\mathbf{w}_{i,i}\cdot, ..., \cdot\mathbf{w}_{n,n} = (l_{i,i}\cdot, ..., \cdot l_{n,n}, D),$$

откуда

$$\mathbf{w}_{i,i} = (l_{i,i}\frac{l_{i+1,i+1}\cdot, ..., \cdot l_{n,n}}{\mathbf{w}_{i+1,i+1}\cdot, ..., \cdot \mathbf{w}_{n,n}}, R) = (l_{i,i}, R) = (a_{i,i}, R).$$

Последнее равенство выполнено потому, что по построению $l_{i,i} \equiv a_{i,i}(modR)$. Таким образом, $w_{i,i} = \mathbf{w}_{i,i}$ вычислен правильно, и по индукции правильно вычислены все диагональные элементы ЭНФ.

Кроме того, из описанного алгоритма приведения к ЭНФ $W_i = \sum c_{i,j}A_j + RB_i$, где $R = D/\mathbf{w}_{i+1,i+1}\cdots\mathbf{w}_{n,n}$, а координаты целочисленного вектора B_i с индексами $i+1, ..., n$ равны нулю. Пусть $\mathtt{A}$ — векторное пространство над Z, порождённое столбцами матрицы A.

Лемма 5.4. $RB_i \in \mathtt{A}$.

Из этой леммы будет следовать, что W_i являются линейными комбинациями столбцов матрицы A, то есть $W = AT$ для некоторой целочисленной матрицы T. Диагональные элементы верхнетреугольной матрицы W совпадают с диагональными элементами $\mathbf{W} = $ЭНФ A. Поэтому совпадают их определители, а значит, $detT = 1$. Согласно алгоритму матрица W имеет ЭНФ, значит она равна $\mathbf{W}$ в силу её единственности.

Доказательство леммы 5.4. Пусть N_i есть $i \times i$-матрица в верхнем левом углу $\mathbf{W}$. Столбцы N_i являются линейными комбинациями столбцов матрицы A, а значит, лежат в $\mathtt{A}, detN_i = \mathbf{w}_{1,1}\cdot, ..., \cdot\mathbf{w}_{i,i}$. По построению $\mathbf{w}_{1,1}\cdot, ..., \cdot\mathbf{w}_{i,i} \mid R$. Пусть $\mathtt{A}_i \subset Z^i \subset Z^n$ - подмодуль, образованный столбцами матрицы N_i. Очевидно, $\mathtt{A}_i \subset \mathtt{A}$, откуда $\mathtt{A}_i \subset \mathtt{A} \cap Z^i, detN_i = [Z^i : \mathtt{A}_i]$. Поскольку порядок элемента делит порядок аддитивной группы $Z^i/\mathtt{A}_i$, то для любого $\bar{z} \in Z^i$ выполнено $\bar{z}detN_i \in \mathtt{A}_i$, откуда $R\bar{z} \in \mathtt{A}_i$. Лемма 5.4 доказана.

Замечание. *Если в рассматриваемом алгоритме убрать редукции* $modR$, *а в шаге 4 вычисление* W_i *заменить на присвоение* $W_i = A_k$ *(при этом будет* $w_{i,i} = d \neq 0$*), убрать переприсвоение при* $a_{i,k} = 0$, *то получится обычный (не модулярный) алгоритм приведения к ЭНФ для матриц, ранг которых совпадает с рангом строк.*

Модулярный алгоритм приведения к СНФ

В данном разделе построен модулярный алгоритм приведения квадратной невырожденной матрицы A, с определителем по модулю равным D к Смитовой нормальной форме V и вычисление унимодулярной матрицы S, для которой существует унимодулярная матрица T такая, что $V = SAT$

Пусть M_i - столбцы, M'_i - строки, а $M_{i,i}$ - подматрица размера $i \times i$ в верхнем левом углу матрицы M. Исходная матрица имеет размеры $n \times n$.

1. $i := n, S := E, R := D$. Если $m = 1$, то $d_1 := R$ и конец.
2. Привести матрицу $A_{i,i}$ к ЭНФ (использовать модулярный алгоритм [49]).
3. $j := i, c := 0$.
4. Если $j = 1$ идти на шаг 6, иначе $j := j - 1$, если $a_{j,i} = 0$, идти на шаг 4.
5. РАЕ $(u, v, d) : ua_{i,i} + va_{j,i} = d = (a_{i,i}, a_{j,i})$.
   $B := uS'_i + vS'_j, S'_j := (a_{i,i}/d)S'_j - (a_{j,i}/d)S'_i, S'_i := B$.
   $B := uA'_i + vA'_j (\mathrm{mod} R), A'_j := (a_{i,i}/d)A'_j - (a_{j,i}/d)A'_i (\mathrm{mod} R), A'_i := B, c := c+1$.
6. Если $c > 0$, идти на шаг 2.
7. Матрица $A_{i,i}$ имеет блочнодиагональный вид (размеры блоков $i-1 \times i-1$ и 1×1) $b := a_{i,i}$. Для всех $0 < k, l < i$ проверка $b \mid a_{k,l}$. Если нет, то $A'_i := A'_i + A'_k, S'_i := S'_i + S'_k$, идти на шаг 2.
8. (Теперь все $a_{k,l}, 0 < k, l \leq i$ делятся на b). Вывести $d_i = (a_{i,i}, R), R := R/d_i$.

9. Если $i = 2$, вывести $d_1 = (a_{1,1}, R)$ и конец, иначе $i := i - 1$, идти на шаг 2.

Обоснование алгоритма приведения к СНФ

Докажем сначала, что в результате алгоритма будет получена $\Delta = \text{СНФ}\, A$. Пусть $\tilde{A}$ - конечная форма матрицы A после применения этого алгоритма. Обозначим $\delta_i(M)$ - НОД $i \times i$ миноров матрицы M. Обозначим ∂_i диагональные элементы Δ. Тогда $\partial_{i+1} \mid \partial_i$ для $i = 1, ..., n-1, D = \partial_1 \cdot, ..., \cdot \partial_n$

$$\partial_i \cdot, ..., \cdot \partial_n = \delta_{n-i+1}(\Delta) = (D, \delta_{n-i+1}(\Delta)) = (D, \delta_{n-i+1}(A)) = (D, \delta_{n-i+1}(L)),$$

где L - диагональная матрица, полученная рассматриваемым алгоритмом при постоянном $R = D$. Предпоследнее равенство верно, поскольку Δ может быть получена из A при помощи элементарных преобразований строк и столбцов, не изменяющих $\delta_k(A)$. Последнее равенство следует из доказательства леммы 5.3, применённого к строкам.

Так как диагональные элементы матрицы L удовлетворяют $l_{i+1} \mid l_i$, то далее с заменой γ на δ, $\mathbf{w}$ на ∂ аналогично последовательно докажем правильность вычисления $\partial_n, \partial_{n-1}, ..., \partial_1$.

Для доказательства корректности вычисления матрицы S заметим, что если убрать редукцию $\bmod R$ в шаге 5, то операции над строками матрицы S совпадают с операциями над строками матрицы A, и поэтому она будет вычислена верно. После преобразований шага 5 алгоритм возвращается на шаг 2, где, согласно алгоритму берётся редукция $(\bmod R)$. Поэтому указанное исключение не повлияет на результаты работы алгоритма.

Замечание. *Если в рассматриваемом алгоритме убрать редукции $(\bmod R)$ и на шаге 8 положить $d_i = a_{i,i}$, то получится обычный алгоритм приведения к СНФ и вычисление матрицы S. Если использовать немодулярный алгоритм приведения к ЭНФ и сопровождать преобразования столбцов матрицы A*

такими же преобразованиями матрицы T *(на старте алгоритма выбрать* $T = E$*), то в используемых обозначениях получим* $\Delta = SAT$*.*

Предложенный в этой статье алгоритм был запрограммирован и использован в комплексе программ, реализующих спуск Вейля для эллиптических кривых над полем характеристики 2.

В работе [67] Сторйохэн предложил модулярный алгоритм вычисления СНФ вместе с матрицей преобразований, работающий в рассматриваемой ситуации за время $O(n^3 \log D)$ без учёта логарифмических множителей вида $\log n$. При этом оценка на память $O(n^2 \log D)$. В этом алгоритме матрицы преобразований строятся после построения СНФ при помощи произведения и обращения нескольких матриц ([67] стр.135-136). В алгоритме левая матрица преобразований строится накапливанием преобразований по ходу построения СНФ. Поэтому необходимая для этого память оценивается величиной $2n^2 \log D$. Для оптимизации времени работы заметим, что переход на шаг 2 в алгоритме происходит при

$$A_{i,i} = \begin{pmatrix} * & * & \cdots & * & & \cdots & & * \\ 0 & * & \cdots & * & & \cdots & & * \\ & \cdots & & & & \cdots & & * \\ 0 & \cdots & 0 & * & * & \cdots & * & 0 \\ 0 & \cdots & 0 & 0 & * & \cdots & * & 0 \\ 0 & \cdots & 0 & & \cdots & & & 0 \\ 0 & \cdots & 0 & 0 & \cdots & 0 & * & 0 \\ 0 & \cdots & 0 & * & * & \cdots & * & a_{i,i} \end{pmatrix} \tag{5.17}$$

Вместо приведения к ЭНФ будем приводить эту матрицу теми же преобразованиями к верхнетреугольному виду (то есть исключим в 4 шаге алгоритма PAE вычисление $W_j = W_j - qW_i$). Кроме того, шаги по j будем проходить в обратном порядке (от 1 до m), что, ввиду специального

вида приводимой матрицы, позволит привести её не более чем за n шагов 3 алгоритма PAE. Это потребует не более, чем $2(n^2 \log D)$ операций, что даёт $2(n^3 \log D)$ для всех i. Это не повлияет на корректность работы алгоритма, так как в доказательстве его корректности использовалось лишь то, что ЭНФ имеет верхнетреугольный вид. Кроме того, в шаге 5 уберём переприсваивание $c := c + 1$ в случае, когда $d = a_{i,i}$ (в этом случае матрица $A_{i,i}$ в шаге 5 остаётся верхнетреугольной). Это приведёт к тому, что переход на шаг 2 из шага 5 при фиксированном i будет осуществляться в среднем не более, чем 2 раза, а ответ "нет"в шаге 7 при больших i не будет реализовываться. Действительно, поскольку вероятность того, что два случайных числа, принадлежащих $\{0, 1, \ldots, \frac{D}{\text{НОД}(\{a_{k,l}\}_{k,l=1,\ldots,i})} - 1\}$ взаимнопросты не меньше $\prod_p (1 - \frac{1}{p^2}) > \frac{1}{2}$, то это будет следовать из предположения о случайности элементов $\{a_{k,l}\}_{k,l=1,\ldots,i}$. Остальная работа в шаге 5 для всех i оценивается величиной $O(n^2 \log D)$.

Таким образом, оценки сложности и памяти получаются такие же как в алгоритме [67], только константы в O представляются меньшими.

5.2 Ядро спуска Вейля

Основные понятия и определения изложены в первом параграфе приложения.

5.2.1 Исследование группового гомоморфизма метода спуска Вейля

В этом подразделе изложены результаты автора из [48]. Метод спуска Вейля в нашем случае заключается в построении гомоморфизма ϕ из группы точек $E(K)$ эллиптической кривой E

$$y^2 + xy + x^3 + \alpha x^2 + \beta = 0, \alpha, \beta \in K \tag{5.18}$$

над большим полем $K = GF(2^{nr})$ характеристики 2 в группу классов дивизоров некоторой гиперэллиптической кривой над относительно меньшим подполем $k = GF(2^r)$ поля K. Этот групповой гомоморфизм является композицией трёх последовательных отображений. Первое из них, ψ, является изоморфизмом $E(K)$ и группы классов дивизоров кривой E:

$$\psi(P_0) = P_0 - P_\infty. \tag{5.19}$$

Второе задаётся функцией $Con_{F/\Upsilon}$, где $F = K(u, v)$ для алгебраических над $\Upsilon = K(x, y)$ элементов u и v, удовлетворяющих следующим уравнениям:

$$x = 1/f(u); \quad y = \sqrt{\beta} + \frac{v}{f^2(u)} \tag{5.20}$$

для некоторого многочлена $f(u) = \lambda_{-1} + \sum_{i=0}^{m-1} \lambda_i u^{2^i}, \lambda_i \in K; \lambda_0 \neq 0, \lambda_{m-1} \neq 0$. Отметим, что возведение в квадрат и извлечение квадратного корня являются невырожденными линейными операторами на K. В новых координатах [48, Часть 1, гл.2, параграф 2] кривая E будет иметь вид

$$v^2 + f(u)v + h(u) = 0, \tag{5.21}$$

где $h(u) = f(u) + \alpha f(u)^2 + \sqrt{\beta} f(u)^3$.

Выясним, из каких точек состоит ядро гомоморфизма, определённого на $E(K)$ формулой

$$\varphi(P) = Con_{F/\Upsilon}(P - P_\infty).$$

Пусть некоторая точка $P(x_0, y_0)$ кривой (5.18) лежит в ядре φ. Тогда, согласно [140, глава IV, §7], следующий дивизор является главным:

$$N_{F/\Upsilon} Con_{F/\Upsilon}(P - P_\infty) = [F : \Upsilon](P - P_\infty) = 2^m P - 2^m P_\infty, m \geq 1. \tag{5.22}$$

Этот дивизор можно получить при помощи m кратного удвоения дивизора $div(x - x_0, y_0)$.

Если $x_0 = 0$, то из уравнения (5.18) $y_0 = \sqrt{\beta}$. Получилась единственная на этой кривой специальная точка, то есть её вторая координата определяется по первой однозначно. Поэтому дивизор $2P - 2P_\infty$ кривой E является главным дивизором, а точнее — дивизором элемента x, а дивизор (5.22) является дивизором элемента $x^{2^{m-1}}$. Это простые следствия определений.

Пусть теперь $x_0 \neq 0$. Применим последовательно алгоритмы сложения и приведения дивизоров для подсчёта приведённого дивизора в классе $2div(x - x_0, y_0)$. Получим:

$$2div(x - x_0, y_0) \sim div((x - x_0)^2, \left[\frac{y_0(x - x_0)}{x_0} + \frac{y_0^2 - x^3 - \alpha x^2 - \beta}{x_0}\right]) =$$

$$div((x - x_0)^2, \left[(\frac{y_0}{x_0} + x_0)x + x_0^2\right]$$

$$\sim div(a_1, b_1),$$

где

$$a_1 = \frac{x^2(\frac{y_0}{x_0} + x_0)^2 + x_0^4 + (\frac{y_0}{x_0} + x_0)x^2 + x_0^2 x + x^3 + \alpha x^2 + \beta}{(x - x_0)^2} =$$

$$x + \frac{\beta}{{x_0}^2} + x_0^2, \quad b_1 \in K.$$

Поскольку род кривой (5.18) равен 1, то получившийся дивизор не является главным, так как согласно [60, Appendix, Теорема 5.1] приведённый вид главного дивизора есть $div(1, 0)$. Поэтому если обозначить $2^i div(x - x_0, y_0) = div(x - x_i, y_i)$, то

$$x_{i+1} = \frac{\beta}{{x_i}^2} + x_i^2. \tag{5.23}$$

Таким образом, дивизор (5.22) будет главным тогда и только тогда, когда для некоторого $j \in \{0, ..., m-1\}, x_j = 0$. В частности, при $j \geq 1$ получим $x_{j-1} = \sqrt[4]{\beta} \in K$ и уравнение

$$y_{j-1}^2 + \sqrt[4]{\beta} y_{j-1} + \sqrt[4]{\beta}^3 + \alpha\sqrt{\beta} + \beta = 0$$

разрешимо в K. Если сделать в этом уравнении замену переменных $y_{j-1} = \sqrt[4]{\beta}(u + \sqrt[4]{\beta})$, то его разрешимость будет равносильна разрешимости уравнения

$$u^2 + u = \alpha, \quad u \in K. \tag{5.24}$$

Поскольку левая часть этого уравнения является линейным оператором с ядром порядка 2, то для половины $\alpha \in K$ оно неразрешимо. Таким образом, справедлива следующая теорема.

Теорема 5.9. *[48] Для кривых вида (5.18), где уравнение (5.24) не разрешимо в K, ядро гомоморфизма φ состоит из точки P_∞ и, возможно, точки $P(0, \sqrt{\beta})$.*

Если уравнение (5.24) разрешимо, то в рассматриваемое ядро, возможно, попадёт ещё точка с координатами $(x_{j-1}, y_{j-1}) = (\sqrt[4]{\beta}, \sqrt[4]{\beta}(u_i + \sqrt[4]{\beta}))$ и сопряжённая к ней $(x_{j-1}, y_{j-1} + x_{j-1})$. При $m \geq 2$ для x_{j-2} получим уравнение

$$\beta + \sqrt[4]{\beta}{x_{j-2}}^2 + x_{j-2}^4 = 0.$$

Замена $x_{j-2}^2 = \sqrt[4]{\beta}z$ показывает, что разрешимость этого уравнения равносильна разрешимости в K уравнения $z^2 + z = \sqrt{\beta}$, что, ввиду того, что возведение в квадрат и извлечение квадратного корня являются невырожденными линейными операторами на K и сохраняют операцию сложения, равносильно разрешимости уравнения

$$z^2 + z = \beta, \quad z \in K. \tag{5.25}$$

Таким образом, для 3/4 всех пар (α, β) из поля K одно из уравнений (5.24) или (5.25) неразрешимо, и в рассматриваемом ядре лежит не более, чем 4 уже рассмотренные точки.

В общем случае точки, которые могут лежать в ядре, могут быть последовательно построены при помощи уравнения (5.23), которое сводится к линейному относительно x_i, и уравнения кривой. Поскольку при фиксированном

x_{i+1} уравнение (5.23), в случае его разрешимости в K, имеет два решения (или одно в случае $x_{i+1} = 0$), то справедлива следующая теорема.

Теорема 5.10. *[48] Ядро гомоморфизма φ содержит не более $1 + 2^{m-1}$ точек.*

Замечание. *Из равенства (5.23) следует, что порядок 2 может быть только у точки $P(0, \sqrt{\beta})$. Так как дивизор функции x равен $2P - 2P_\infty$ и отображение (5.19) является изоморфизмом, эта точка действительно имеет порядок 2.*

Лемма 5.5. *[48] Пусть $P_0 = P(x_0, y_0)$ - конечная точка кривой $E(K)$, тогда $Con_{F/\Upsilon}(P_0 - P_\infty) = div(f(u) + \frac{x_0}{\sqrt{\beta}}, \frac{x_0+y_0+\sqrt{\beta}}{\sqrt{\beta}})$*

Доказательство. Многочлен $f(u)$ не имеет кратных корней, поэтому индекс ветвления P равен 1. Легко понять, что

$$Con_{F/\Upsilon}(P_0) = div(1 - x_0 f(u), v_0), v_0 = \frac{\sqrt{\beta} + y_0}{x_0^2}.$$

Вычислим теперь $Con_{F/\Upsilon}(P_\infty)$. Если P - точка поля F, лежащая над P_∞, то она лежит и над бесконечной точкой поля $K(x)$, то есть

$$\frac{1}{x} = f(u) \in P \text{ как в идеале.}$$

Отсюда следует, что P - конечная точка поля F. Равенство (5.21) и включение $f(u) \in P$ означают, что $v^2 \in P$, и так как P - простой идеал, заключаем, что $v \in P$. Таким образом, справедливо включение $(f(u), v) \subset P$.

Обратно, если P - точка поля F, порождённая простым идеалом кольца $K[u, v]$, содержащим идеал $(f(u), v) \subset K[u, v]$, то $\frac{1}{x} = f(u) \in P$, и, значит, $\frac{1}{x} \in P \cap E(K)$. Согласно [143, Предложение III.1.4, глава 3], пересечение $p = P \cap E(K)$ есть точка поля $E(K)$. Поскольку она содержит $\frac{1}{x}$, то она является бесконечно удалённой точкой поля $E(K)$, то есть P_∞ в силу её единственности, что следует из леммы 1.3.

Поэтому образ $P_0 - P_\infty$ после применения конормы будет классом дивизоров кривой (5.21) над полем F с представителем

$$div(1 - x_0 f(u), v_0) - div(f(u), 0). \tag{5.26}$$

Поскольку $div(f(u)) = 2div(f(u), 0)$, то дивизор (5.26) можно представить в виде

$$= div(1 - x_0 f(u), v_0) + div(f(u), 0).$$

Применим к этому выражению алгоритм сложения классов дивизоров (см. Приложение гл.1). Последовательно получим $l_1 = 1, l = 1, e_1 = 1, e_2 = x_0, g_1 = 1, g_2 = 0 = w$. Результат сложения можно представить следующим образом:

$$div(f(u)(1 - x_0 f(u)), [f(u)(1 - x_0 f(u)) + x_0 f(u) v_0]) =$$

$$= div(f(u)(1 - x_0 f(u)), x_0 v_0 f(u)).$$

Последнее равенство справедливо, потому что взятие второй компоненты по модулю первой не меняет класс дивизоров. Род кривой (5.21) - g, согласно [69, Лемма 9], равен 2^{m-1} или $2^{m-1} - 1$. Поэтому полученный в результате сложения полуприведённый дивизор, не является приведённым. Хотя запись (5.21) не является каноническим видом кривой, к ней можно применять стандартный алгоритм приведения. Результат его работы будет лежать в том же классе, однако не обязательно будет приведённым дивизором. Применив один шаг алгоритма (см. гл. 1), получим:

$$a_1(u) = \frac{(x_0 v_0 f(u))^2 + x_0 v_0 f(u)^2 + h}{f(u)(1 - x_0 f(u))} =$$

$$\frac{f(u)((x_0 v_0)^2 + x_0 v_0) + 1 + \alpha f(u) + \sqrt{\beta} f(u)^2}{1 - x_0 f(u)} =$$

$$\frac{1 + ((x_0 v_0)^2 + x_0 v_0 + \alpha + \frac{\sqrt{\beta}}{x_0}) f(u) - f(u) \frac{\sqrt{\beta}}{x_0} (1 - x_0 f(u))}{1 - x_0 f(u)} =$$

$$\frac{\sqrt{\beta}}{x_0} f(u) + \frac{1 + f(u)(\alpha + \frac{y_0 + \sqrt{\beta}}{x_0} + \frac{y_0^2 + \beta}{x_0^2} + \frac{\sqrt{\beta}}{x_0})}{1 - x_0 f(u)} = \frac{\sqrt{\beta}}{x_0} f(u) + 1$$

$$b_1(u) \equiv (1 + x_0 v_0) f(u) \equiv (1 + \frac{y_0 + \sqrt{\beta}}{x_0}) \frac{x_0}{\sqrt{\beta}} \pmod{a_1(u)},$$

или

$$b_1(u) \equiv \frac{x_0 + y_0 + \sqrt{\beta}}{\sqrt{\beta}} \pmod{a_1(u)}.$$

Следовательно, образ отображения $Con_{F/\Upsilon}$ примет вид:

$$div(f(u) + \frac{x_0}{\sqrt{\beta}}, y - \frac{x_0 + y_0 + \sqrt{\beta}}{\sqrt{\beta}}). \tag{5.27}$$

Лемма 5.5 доказана.

Заметим, что в случае, если $g = 2^{m-1}$, этот дивизор является нетривиальным приведённым дивизором для любой конечной точки P_0.

Спуск Вейля завершает следующая замена переменных [70, (2.14), §2, гл.1, часть 1]

$$\begin{cases} \widetilde{u} = \lambda_0 u + \lambda'' \\ \widetilde{v} = v + g(\widetilde{u}) t(\widetilde{u}), \end{cases}$$

где $g(\widetilde{u}) = f(\frac{\widetilde{u} - \lambda''}{\lambda_0})$, а $\lambda'' \in K$ выбрано так, что $g(\widetilde{u}) \in k[\widetilde{u}]$, а затем взятие нормы.

В новых переменных кривая (5.21) запишется следующим уравнением над k:

$$\widetilde{v}^2 + g(\widetilde{u})\widetilde{v} + g(\widetilde{u})(1 + ag(\widetilde{u}) + bg(\widetilde{u})^2) = 0, \text{где } a, b \in k. \tag{5.28}$$

Дивизор (5.27) в новых переменных запишется следующим образом:

$$\begin{gathered} div(g(\widetilde{u}) + \tfrac{x_0}{\sqrt{\beta}}, \left[\tfrac{x_0 + y_0 + \sqrt{\beta}}{\sqrt{\beta}} + g(\widetilde{u}) t(\widetilde{u})\right]) = \\ div(g(\widetilde{u}) + \tfrac{x_0}{\sqrt{\beta}}, \left[\tfrac{x_0 + y_0 + \sqrt{\beta}}{\sqrt{\beta}} + \tfrac{x_0}{\sqrt{\beta}} t(\widetilde{u})\right]), \end{gathered} \tag{5.29}$$

Поскольку многочлен g получен из многочлена f при помощи линейной замены переменных, то коэффициент в одночлене первой степени у него умножится на ненулевую константу (легко увидеть, что это будет 1), а все

остальные одночлены будут иметь степени, равные степеням двойки. Поэтому, многочлен $g(\widetilde{u}) + \frac{x_0}{\sqrt{\beta}}$ не имеет кратных корней.

Обозначим $f_u(x)$ минимальный многочлен элемента u над k.

Теорема 5.11. *Пусть n - простое нечётное число и $\frac{x_0}{\sqrt{\beta}} \notin k$. Тогда для построенного методом спуска Вейля гомоморфизма ϕ выполнено: $\phi(P(x_0, y_0)) = div(f_{\frac{x_0}{\sqrt{\beta}}}(g(\widetilde{u})), G(\widetilde{u}))$, для некоторого $G(\widetilde{u}) \in k[\widetilde{u}]$. Если $P_0 = P(x_0, y_0), P_1 = P(x_1, y_1)$ и для некоторого $i \in \{0, 1, ..., n-1\}$, выполнено: $x_1 = \sqrt{\beta}\sigma^i \frac{x_0}{\sqrt{\beta}}$, тогда $\phi(P_0) = \phi(P_1)$.*

Доказательство. По условию теоремы $deg_k \frac{x_0}{\sqrt{\beta}} = n$. Обозначим $u_1, ..., u_{2^{m-1}}$ — однократные корни многочлена $g(\widetilde{u}) + \frac{x_0}{\sqrt{\beta}}$. Корни сопряжённого к нему многочлена $\sigma(g(\widetilde{u}) + \frac{x_0}{\sqrt{\beta}}) = g(\widetilde{u}) + \sigma(\frac{x_0}{\sqrt{\beta}})$, где σ - некоторое произвольное продолжение автоморфизма $\sigma : \sigma w = w^{2^r}, w \in K$ на $\overline{K}$, имеют вид $\sigma u_1, ..., \sigma u_{2^{m-1}}$. В рассматриваемом случае $\sigma^j(g(\widetilde{u}) + \frac{x_0}{\sqrt{\beta}}) - (g(\widetilde{u}) + \frac{x_0}{\sqrt{\beta}}) = \sigma^j(\frac{x_0}{\sqrt{\beta}}) - \frac{x_0}{\sqrt{\beta}}$, поэтому $\sigma^j u_i \notin \{u_1, ..., u_{2^{m-1}}\}$ для любых $i \in 1, ..., 2^{m-1}$, при $n \nmid j$. Это также означает, что минимальные многочлены элементов u_i над k различны. Дивизор (5.29) имеет вид $\sum_{i=1}^{2^{m-1}} div(\widetilde{u} - u_i, v_i); u_i, v_i \in \overline{K}$, а его норма над $k[\widetilde{u}, \widetilde{v}]$ есть

$$\sum_{j=1}^{n} \sum_{i=1}^{2^{m-1}} div(\widetilde{u} - \sigma^j u_i, \sigma^j v_i), \tag{5.30}$$

(См. [130, Следствие 3 Предложения 2.1, глава 1]). Все точки этого дивизора однократны, их первые координаты различны, поэтому он полуприведённый. Значит, существует многочлен $G(x) \in k[x]$ такой, что этот дивизор имеет вид

$$div(N(g(\widetilde{u}) + \frac{x_0}{\sqrt{\beta}}), G(x)). \tag{5.31}$$

Многочлен G может быть найден при помощи метода неопределённых коэффициентов из уравнений $G(u_i) = \frac{x_0 + y_0 + \sqrt{\beta}}{\sqrt{\beta}} + \frac{x_0}{\sqrt{\beta}} t(u_i), \quad i = 1, ..., 2^{m-1}$.

Поскольку коэффициенты многочлена g лежат в k, то

$$N(g(\widetilde{u})+\frac{x_0}{\sqrt{\beta}})=\prod_{j=1}^{n}(g(\widetilde{u})+\sigma^j\frac{x_0}{\sqrt{\beta}})=f_{\frac{x_0}{\sqrt{\beta}}}(g(\widetilde{u}))=\prod_{i=1}^{2^{m-1}}f_{u_i}(\widetilde{u}),$$

где $f_{\frac{x_0}{\sqrt{\beta}}}(z)\in k[z]$ - минимальный многочлен элемента $\frac{x_0}{\sqrt{\beta}}\in K$, а в последнем произведении стоят минимальные многочлены элементов $u_1,...,u_{2^{m-1}}$, которые лежат в $k[x]$, неприводимы и в нашем случае различны. Значит, дивизор (5.31) является суммой 2^{m-1} точек поля $k(\widetilde{u},\widetilde{v})$ вида $div(f_{u_i}(\widetilde{u}),G_i(\widetilde{u}))$.

Отметим, что образ точки $P_1=P(x_1,y_1)$, где для некоторого i выполнено: $\sigma^i\frac{x_0}{\sqrt{\beta}}=\frac{x_1}{\sqrt{\beta}}$, может отличаться от полученного образа точки P_0 только вторыми координатами: пусть это будут многочлены $G'_i(\widetilde{u})\in k[\widetilde{u}]$, которые в рассматриваемом случае, также как и G_i, являются решениями сравнений

$$G'^{\,2}_i+gG'_i+g(1+ag+bg^2)\equiv 0\pmod{f_{u_i}}.$$

Отсюда

$$(G_i-G'_i)^2+g(G_i-G'_i)\equiv 0\pmod{f_{u_i}}.$$

Поэтому, ввиду неприводимости над k многочленов f_{u_i} и того, что $degG_i, degG'_i<degf_{u_i}$, имеем $G'_i\in\{G_i,G_i+g\}$. Таким образом, точки в образе P_1 либо совпадают с точками образа P_0, либо совпадают с сопряжёнными к ним, а поэтому в классе дивизоров получим тот же образ. Теорема 5.12 доказана.

Пусть теперь n - нечётное и $\frac{x_0}{\sqrt{\beta}}\in k$ (например, $(x_0,y_0)=(0,\sqrt{\beta})$) тогда уравнение (12) перепишем в виде

$$\left(\frac{y_0}{\sqrt{\beta}}\right)^2+\frac{y_0}{\sqrt{\beta}}\frac{x_0}{\sqrt{\beta}}+1+\alpha\left(\frac{x_0}{\sqrt{\beta}}\right)^2+\sqrt{\beta}\left(\frac{x_0}{\sqrt{\beta}}\right)^3=0,$$

откуда для $v=Tr_{K/k}(\frac{y_0}{\sqrt{\beta}})+1$ имеем

$$v^2+v\frac{x_0}{\sqrt{\beta}}+\frac{x_0}{\sqrt{\beta}}+Tr_{K/k}(\alpha)\left(\frac{x_0}{\sqrt{\beta}}\right)^2+Tr_{K/k}(\sqrt{\beta})\left(\frac{x_0}{\sqrt{\beta}}\right)^3=0.$$

Значит, (см. уравнение (5.28)) вторые координаты точек дивизора (5.31) принадлежат k и равны $Tr_{K/k}(\frac{y_0}{\sqrt{\beta}})+1$, или $Tr_{K/k}(\frac{y_0}{\sqrt{\beta}})+1+\frac{x_0}{\sqrt{\beta}}$. В частности, при $x_0 = 0$ они совпадают и равны нулю. В то же время, σ осуществляет перестановку их первых координат, поэтому рассматриваемая норма (5.30) при $x_0 = 0$ равна

$$n\sum_{i=1}^{2^{m-1}} div(\widetilde{u} - u_i, 0).$$

Поскольку все точки в этой сумме специальные, а n - нечётное, то этот дивизор эквивалентен $\sum_{i=1}^{2^{m-1}} div(\widetilde{u} - u_i, 0)$ (переход осуществляется дивизором элемента $g(u)^{\frac{n-1}{2}}$). Согласно [69], если род кривой (5.28) не равен $2^{m-1} - 1$, то он равен 2^{m-1}, и этот дивизор является приведённым, а значит, не является главным, то есть не лежит в ядре.

Если исходная точка P_0 имела чётный порядок $2^r s$, s - нечётно, $r \geq 1$, то $2^{r-1}sP_0$ - точка порядка 2. Поэтому, согласно приведённому выше замечанию, — это точка с координатами $(0, \sqrt{\beta})$, которая по доказанному не лежит в ядре. Значит, степень вхождения двойки в порядок образа точки P_0 равна r.

Суммируя сказанное, получим следующую теорему.

Теорема 5.12. *[48] Пусть n - нечётное число, и род кривой (5.28) не равен $2^{m-1} - 1$. Тогда для построенного методом спуска Вейля гомоморфизма ϕ выполнено:*

$P(0, \sqrt{\beta}) \notin Ker\phi$, и степени вхождения двойки в порядки точек кривой E и их образов совпадают.

Глава 6

Решение разреженных систем линейных уравнений над $GF(2)$

При атаках на системы защиты информации, основанные на сложности задачи факторизации целых рациональных чисел, возникает необходимость решать большие системы разреженных линейных уравнений над полем $\mathbb{F} = GF(2)$ [144]. При этом обычно не требуется искать все решения, ограничиваясь нахождением нескольких, отличных от нулевого в случае однородной системы. Разреженность означает, что число ненулевых членов в каждом уравнении рассматриваемой системы не превосходит некоторой сравнительно небольшой величины d, в то время как число переменных системы N рассматривается как растущий параметр. Для задач факторизации, решаемых на современной компьютерной технике, общее число переменных оценивается несколькими сотнями миллионов, в то время как параметр d как правило не превосходит 1000.

На практике, ввиду особенностей работы с полем $\mathbb{F}$, вместо векторов удобно рассматривать блоки, то есть матрицы из $\mathbb{F}^{N\times n}$, где n – блочный параметр, обычно равный, или кратный, длине машинного слова (32, 64, или 128). Дело в том, что современные процессоры за один такт своей работы обрабатывают однобитное и 64-битное число (машинное слово) за одинаковое время. Есть процессоры, обрабатывающие в одном такте сразу несколько

машинных слов. Алгоритмы, работающие с машинными словами, называются блочными алгоритмами.

Напомним, что пространством Крылова матрицы A с начальным блоком B называется пространство, образованное столбцами $A^iB, i = 0, 1, \ldots$.

Блочные алгоритмы решения разреженных линейных систем основаны на одной из двух идей, которые сводят вычисление решения к многократному умножению на матрицу системы, которое, ввиду разреженности последней, занимает мало компьютерного времени. Первое направление - это алгоритмы, построенные на основе идеи Ланцоша (C. Lanczos) о последовательной ортогонализации пространства Крылова. Блочная реализация этого подхода называется алгоритмом Монтгомери (P.Montgomery) [145]. Второе направление - алгоритмы на основе идеи Видемана (D.H. Wiedemann), построения приближения формальных рядов порядка, сравнимого с размерностью матрицы, что фактически приводит к построению характеристического многочлена матрицы, из которого и получается формула для решения соответствующей линейной системы уравнений. Блочный вариант этого алгоритма построен Копперсмитом (D.Coppersmith) [146].

Полное описание алгоритма факторизации и алгоритмов решения разреженных линейных систем занимает очень много места, и мы ограничимся в Приложении некоторым переложением на удобный нам язык блочных алгоритмов Ланцоша-Монтгомери [145] и Видемана-Копперсмита [146]. Остальные подробности можно посмотреть в книге Василенко О.Н. [95].

В данной главе предложен новый блочный алгоритм решения разреженных систем линейных уравнений, построенный на свойствах преобразования приближений формальных рядов в ортогональный базис пространства Крылова. Данный подход позволяет связать, в каком-то смысле, идеи Ланцоша и Видемана. Побочным результатом стал новый алгоритм построения приближений Паде к рядам с матричными коэффициентами.

Далее последовательно излагаются несколько версий нового алгоритма. Для наилучшей с нашей точки зрения, оптимизированной версии доказана оценка сложности, представленная в теореме **??**.

6.1 Замечания о симметрических матрицах

Пусть $N, M \in \mathbb{N}$, и требуется найти нетривиальное решение системы линейных однородных уравнений вида

$$DX = 0, D \in \mathbb{F}^{M \times N}, X \in \mathbb{F}^{N \times n}, M < N, \tag{6.1}$$

где n – блочный параметр, обычно равный длине машинного слова, 32 или 64 .

Выберем случайно блок $Y \in \mathbb{F}^{N \times n}$ и рассмотрим систему

$$AX = B, A \in \mathbb{F}^{N \times N}, B, X \in \mathbb{F}^{N \times n}, \tag{6.2}$$

где $A = D^T D \in \mathbb{F}^{N \times N}$, в этих условиях вырожденная симметричная матрица, $B = AY$. Заметим, что любое решение X_D системы (6.1) позволяет построить X_A - решение системы (6.2) по формуле

$$X_A = X_D + Y.$$

Обратно, если есть X_A - решение системы (6.2), то $D^T D(X_A - Y) = 0$. Если $rang D = M$, то из полученного равенства находим, что $D(X_A - Y) = 0$. То есть $X_A - Y$ - решение системы (6.1).

Обозначим $dim_{(6.1)} X$ размерность пространства, полученого в пересечении пространства $\langle X \rangle$, образованного столбцами произвольного блока X, с ядром линейного оператора D. Обозначим $dim_{(6.2)} X$ размерность пространства, полученого в пересечении пространства, образованного столбцами блока X, с ядром линейного оператора $A = D^T D$, а $dim'_{(6.2)} X$ размерность пространства, полученого в пересечении пространства, образованного столбцами блока X, с ядром линейного оператора $A' = DD^T$.

Теорема 6.1. *[75]*

$$1) dim_{(6.1)} X \geq dim_{(6.2)} X - (M - rangD),$$

$$2) dim_{(6.1)} D^T X \geq dim'_{(6.2)} X - (M - rangD).$$

Доказательство. 1) В соответствии с размерами матрицы D имеем $dimKerD^T = M - rangD$. По построению вектор x лежит в $KerA$ тогда и только тогда, когда он лежит в $KerD$ либо $Dx \in KerD^T \setminus \{0\}$. Количество линейно независимых по модулю $KerD$ векторов, удовлетворяющих второму из этих условий не превосходит $dimKerD^T$. Другими словами, если дополнить базис $KerD$ до базиса $KerA$, то векторов из дополнения, по теореме о гомоморфизме, будет не более, чем размерность образа. То есть $dimKerD^T$. Поэтому коранг $KerD$ в $KerA$ не больше $M - rangD$. Пересекая с $\langle X \rangle$, получим доказываемое утверждение.

Неравенство 2) эквивалентно неравенству

$$dimD^T X_1 \geq dimX_1 - (M - rangD),$$

где X_1 выбрано так, чтобы $\langle X_1 \rangle = \langle X \rangle \cap KerA'$. Количество линейно независимых векторов среди столбцов блока $D^T X$, уменьшится по сравнению с количеством линейно независимых векторов среди столбцов блока X не более, чем на $dimKerD^T = M - rangD$. Это и есть доказываемое неравенство.

Теорема доказана.

Отметим также, что при $n - 1 \geq rangD - rangD^T D$ коразмерность пространства решений однородной системы с матрицей D, вложенного в пространство решений однородной системы с матрицей $D^T D$, не больше, чем $n - 1$. Поэтому если нам удалось построить n линейно независимых решений второй из этих систем, то некоторая их нетривиальная линейная комбинация будет решением первой.

Далее будем предполагать, что A — произвольная симметричная матрица.

Определим A-скалярное произведение двух векторов $x, y \in \mathbb{F}^N$ как $(x, y) = x^T Ay$. Обозначим, как и раньше, $\langle W \rangle$ векторное подпространство в $\mathbb{F}^N$, порождённое столбцами матрицы W. Будем говорить, что пространства $\langle V \rangle$ и $\langle W \rangle$ A-ортогональны, если матрица $V^T AW$ попарных A-скалярных произведений, образующих их векторов равна нулю. Если матрица $W^T AW$ обратима, то будем говорить, что A-скалярное произведение на пространстве $\langle W \rangle$ невырождено. Обозначим также I_n единичную, а O_n - нулевую матрицы размера $n \times n$.

Теорема 6.2. *[72] Пусть матрица $W \in \mathbb{F}^{N \times M}$ такова, что*

$$W^T AW - \textit{невырожденная матрица,} \tag{6.3}$$

а $u \in \mathbb{F}^N$. Тогда существует единственный вектор $w \in \langle W \rangle$, для которого

$$W^T A(u - w) = 0, \tag{6.4}$$

при этом

$$w = W(W^T AW)^{-1} W^T Au. \tag{6.5}$$

Доказательство.

Существование. Пусть w определён формулой (6.5), тогда $W^T A(u - w) = W^T A(u) - W^T A(u) = 0$.

Единственность. Пусть выполнено равенство (6.4). Поскольку $w \in \langle W \rangle$, то $w = Wc$ для некоторого вектора c, откуда

$$W^T Au = W^T AWc,$$

или

$$c = (W^T AW)^{-1} W^T Au,$$

то есть выполнено равенство (6.5). Теорема доказана.

Будем в дальнейшем называть w из теоремы 6.2 A-ортогональной проекцией вектора u на линейное пространство $\langle W \rangle$.

Если теперь предположить, что u — решение уравнения (6.2), то его A-ортогональная проекция на любое подпространство $\langle W \rangle$, базис которого, записанный по столбцам, образует матрицу W, удовлетворяющую условию (6.3), может быть вычислена однозначно по формуле

$$w = W(W^T AW)^{-1} W^T B. \tag{6.6}$$

Если же это решение лежит в $\langle W \rangle$, то в $\langle W \rangle$ оно единственно и определяется формулой (6.6).

Теорема 6.3. *[72] Пусть имеется набор попарно A-ортогональных подпространств $\langle W_i \rangle, i = 0, 1, \ldots, m-1$ в $\mathbb{F}^N$, A-скалярное произведение на которых невырождено. Пусть также матрица A переводит сумму этих подпространств $\langle W \rangle$ в себя и $\langle B \rangle \subseteq \langle W \rangle$. Тогда решение системы (6.2) в $\langle W \rangle$ единственно и задаётся формулой*

$$X = \sum_{i=0}^{m-1} W_i ({W_i}^T A W_i)^{-1} {W_i}^T B. \tag{6.7}$$

Доказательство.

Как показано в теореме 6.2 при выполнении условий теоремы, формула (6.7) однозначно определяет A-ортогональную проекцию решения системы (6.2) на пространство $\langle W \rangle$. Для окончательного доказательства осталось показать, что эта формула задаёт решение системы (6.2). Легко понять, что ${W_j}^T AX = {W_j}^T B, j = 0, 1, \ldots, m-1$, поэтому все вектора блока $AX - B$ A-ортогональны пространству $\langle W \rangle$. Кроме того, так как по условию $\langle B \rangle \subseteq \langle W \rangle$ и $A\langle W \rangle \subseteq \langle W \rangle$, то $\langle AX - B \rangle \subseteq \langle W \rangle$, откуда, ввиду невырожденности A-скалярного произведения на $\langle W \rangle$, имеем $AX - B = 0$. Теорема доказана.

Рассмотрим общую ситуацию.

Теорема 6.4. *[75] Пусть пространство Крылова* $\langle W\rangle$ *построено в виде суммы непересекающихся* A*-ортогональных пространств* $\langle W_i\rangle, i = 0, 1, \ldots, m$ *с начальным блоком вида* $W_0 = B = AY$*, на первых* m*, из которых* A*-скалярное произведение невырождено. Пусть* $\langle W_m\rangle$ A*-ортогонально всему* $\langle W\rangle$*, в частности себе. Пусть вычислено* AW_m *и* X *по формуле (6.7),* $dim\langle W_m\rangle = \mu > 0$*. Тогда не более чем за* $2\frac{(n+\mu-1)(n+\mu)}{2}N$ *битовых операций с вероятностью не меньше* $1 - \frac{1}{2^n}$ *(при условии статистической независимости* $\langle Y\rangle$ *и* $\langle W\rangle$*) может быть вычислено решение системы (6.2).*

Доказательство. Заметим, что по условию пространство $\langle W_m\rangle$ A-ортогонально пространству $W = \langle W_0\rangle + \cdots + \langle W_m\rangle$, в частности, A-ортогонально себе. Кстати заметим, что на практике размерность $\langle W_m\rangle$ небольшая (скажем 2). Также по условию $\langle W_m\rangle$ содержит все вектора пространства Крылова, ортогональные всему этому пространству. Тогда AW_m имеет вид $w_0 + w_1 + \cdots + w_m, \langle w_i\rangle \subseteq \langle W_i\rangle, i = 1, 2, \ldots, m$. Получаем при $j \in \{1, \ldots, m-1\}$ цепочку равенств

$$w_j{}^T AW_j = (w_0 + w_1 + \cdots + w_m)^T AW_j =$$
$$(AW_m)^T AW_j = (W_m)^T A(AW_j) \subseteq (W_m)^T AW = \{0\},$$

которая, ввиду невырожденности A-скалярного произведения на W_j, показывает, что $w_j = 0$ при $j = 0, 1, \ldots, m-1$, то есть $AW_m \subseteq \langle W_m\rangle$.

Если $\langle AW_m\rangle \neq \langle W_m\rangle$, то элемент из ядра матрицы A может быть получен приведением к нижнетреугольному виду правой части равенства $AW_m = Z$ за $2\frac{(\mu-1)\mu}{2}N$ операций.

Пусть теперь $\langle AW_m\rangle = \langle W_m\rangle$. Из формулы (6.7), как при доказательстве теоремы 6.3, подстановкой получим

$$(W_j)^T(AX - B) = (W_j)^T B - (W_j)^T B = 0, j = 0, 1, ..., m-1,$$
$$0 = \langle (W_m)^T A(AX - B)\rangle = \langle (W_m)^T (AX - B)\rangle,$$

так как $AX-B \subseteq W$. Поэтому, ввиду A-инвариантности пространства Крылова,

$$(W_j)^T A(AX-B) = (AW_j)^T(AX-B) \subseteq W^T(AX-B) = \{0\}, j = 0, 1, ..., m-1,$$

то есть вектора блока $AX-B$ A-ортогональны W, то есть $\langle AX-B\rangle \subseteq \langle W_m\rangle$. Значит, линейный оператор A отображает сумму пространств $\langle X-Y\rangle + \langle W_m\rangle$ в $\langle W_m\rangle$. Если $\langle Y\rangle \not\subseteq \langle W\rangle$, то размерность первого пространства больше, чем размерность второго, и при помощи Гауссовых исключений в равенстве $A(X-Y\|W_m) = Z, \langle Z\rangle \subseteq \langle W_m\rangle$ можно найти элемент из ядра A.

Вероятность того, что $\langle Y\rangle \not\subseteq \langle W\rangle$ при условии статистической независимости этих пространств и вырожденности матрицы A, откуда $\langle W\rangle \neq \mathbb{F}^N$, очевидно, оценивается величиной $1-\frac{1}{2^n}$, где n - число столбцов в блоке Y. Сложность одной элементарной операции со столбцами матриц $X-Y\|W_m, Z \in \mathbb{F}^{N\times(n+\mu)}$ оценивается величиной N, а число таких операций - величиной $2((n+\mu-1)+...+1) = 2^{\frac{(n+\mu)(n+\mu-1)}{2}}$.

Теорема доказана.

Отметим, что условие $W_0 = B$ в последней теореме может быть заменено на $W_0 = Y$. Тогда аналогично получим $A(X-Y) \subseteq \langle W_m\rangle$, причём всегда $\langle X-Y\rangle \subseteq \langle W\rangle \setminus \langle W_m\rangle$. Правда в этом случае $\langle Y\rangle \subseteq \langle W\rangle$ и мы не сможем оценить вероятность получения решения как это было сделано выше.

Некоторые похожие утверждения использованы в [145] без формулировок и доказательств.

6.2 Блочные алгоритмы

6.2.1 Новый алгоритм типа Ланцоша-Паде (первая версия)

Про матрицу A системы мы будем предполагать следующее: интересующий нас размер имеет порядок $N \sim M \sim 10^8$; среднее число единиц в строке матрицы не велико и, скажем, не превосходит 10^3; ненулевые элементы матрицы

располагаются существенно стохастически, то есть их местоположение заранее определено быть не может.

Среди методов, которые наиболее часто упоминаются в связи с задачей решения разреженных систем линейных уравнений, для нас будут особенно важны блочный метод Ланцоша в изложении Монтгомери [145] и блочный алгоритм Видемана в изложении Копперсмита [146]. Мы интересуемся этими методами сразу по двум причинам. Во-первых, предлагаемый нами алгоритм наследует (как мы надеемся) свойства обоих алгоритмов. А, во-вторых, реализация этих алгоритмов на распределенных вычислительных системах имеет существенные отличия.

Прежде чем переходить к дальнейшему изложению, необходимо сказать несколько слов о том, какие вычислительные системы мы имеем ввиду при реализации алгоритма. Во-первых, мы ориентируемся на системы с распределенной памятью, то есть на тот случай, когда каждому вычислительному узлу соответствует свой банк памяти, а обмен данными между узлами происходит по внешним линиям связи. Во-вторых, мы ничего не будем предполагать о топологии линий связи, то есть о том, какие вычислительные узлы соединены линиями связи, а какие нет. Более того, исследуя алгоритмы, мы часто будем предполагать, что такая топология имеет простейший вид кольца, когда каждый вычислительный модуль соединен с двумя другими модулями, называемыми его соседями, а все соединения образуют цепочку, причем все линии цепи имеют одинаковую пропускную способность. Как показывает практика работы на вычислительных системах с многопользовательским режимом, такое предположение едва ли является слишком грубым.

Цель этого подраздела — неформально представить идеи, лежащие в основе предлагаемого алгоритма.

Использование блочного фактора s, то есть блоков не равного, а кратного длине машинного слова с коэффициентом s, для улучшения параллельных

свойств метода Монтгомери сопряжено с необходимостью для процессоров обмениваться большими объемами данных, причем время, затрачиваемое на такие обмены в версии [158] столь велико, что не позволяет надеяться на существенное увеличение скорости решения задач. Напротив, использование блочного фактора в алгоритме Видемана-Копперсмита является решающим в построении параллельного алгоритма. По нашему мнению основная причина лучших параллельных свойств метода Копперсмита, по сравнению с методом Монтгомери, состоит в отсутствии выражений вида V^TAV с блоками V размера $N \times 64s$, которые меняются от шага к шагу алгоритма.

Наша цель — построить алгоритм, в котором операции вида V^TAV, с переменными блоками V, либо отсутствуют, либо существенно сокращаются. Кроме того, новый метод должен обладать простыми, аналогичными методу Монтгомери, вычислениями решения системы X.

Скажем заранее несколько слов об устройстве алгоритма. Сначала рассматривается метод построения блочного квази A - ортогонального базиса пространства Крылова на основе аппроксимаций типа Паде, за счет того, что при построении коэффициентов ряда (имеется аналогия с методом Копперсмита) удобно использовать параллельный алгоритм умножения матрицы на вектор с блочным фактором s. Затем мы вводим процедуру «доортогонализации», с помощью которой из квази A - ортогонального базиса пространства Крылова получаются A - ортогональные блоки, с помощью которых строится решение системы.

Пусть задана система уранений над полем $\mathbb{F}$

$$AX = B \tag{6.8}$$

с симметричной матрицей A размера $N \times N$ и с блочными векторами B и X, размер которых $N \times 64s$.

По аналогии с методом Копперсмита рассмотрим бесконечный матричный ряд $\alpha(\lambda)$

$$\alpha(\lambda) = \sum_{i=0}^{\infty} \alpha_i \lambda^{-i} \tag{6.9}$$

с коэффициентами $\alpha_i = B^T A^i B$. В отличие от Копперсмита мы будем рассматривать ряд по отрицательным степеням формальной переменной.

Замечание. *Мы везде далее полагаем, что организация вычислений коэффициентов α_i полностью аналогична той, что используется в методе Копперсмита, а именно, вектор B вида*

$$B = \left[\begin{array}{cccc} B_1 & B_2 & \cdots & B_s \end{array} \right],$$

хранится на каждом вычислительном узле полностью. Узел с номером j вычисляет, во-первых, векторы $A^i B_j$, а, во-вторых, части α_{ij} матриц α_i следующего вида

$$\alpha_{ij} = B^T A^i B_j.$$

Используя коэффициенты ряда (6.9), будем последовательно строить матричные приближения типа Паде двух видов

$$\alpha(\lambda) Q^{(s)}(\lambda) - P^{(s)}(\lambda) = \sum_{i=s+1}^{\infty} \alpha_i^{(s)} \lambda^{-i} = \alpha^{(s)}(\lambda), \tag{6.10}$$

$$\alpha(\lambda) \tilde{Q}^{(s)}(\lambda) - \tilde{P}^{(s)}(\lambda) = \sum_{i=s-\delta(s)}^{\infty} \tilde{\alpha}_i^{(s)} \lambda^{-i} = \tilde{\alpha}^{(s)}(\lambda), \tag{6.11}$$

здесь $Q^{(s)}(\lambda)$, $P^{(s)}(\lambda)$, $\tilde{Q}^{(s)}(\lambda)$ и $\tilde{P}^{(s)}(\lambda)$ – матричные полиномы от λ степени не выше s, коэффициенты которых — квадратные матрицы размера $64s \times 64s$, $\delta(s) \in \mathbb{N} \cup \{0, -1\}$

Пусть

$$
\begin{aligned}
Q^{(s)}(\lambda) &= Q_s^{(s)}\lambda^s + Q_{s-1}^{(s)}\lambda^{s-1} + \cdots + Q_1^{(s)}\lambda + Q_0^{(s)}, \\
P^{(s)}(\lambda) &= P_s^{(s)}\lambda^s + P_{s-1}^{(s)}\lambda^{s-1} + \cdots + P_1^{(s)}\lambda + P_0^{(s)}, \\
\tilde{Q}^{(s)}(\lambda) &= \tilde{Q}_s^{(s)}\lambda^s + \tilde{Q}_{s-1}^{(s)}\lambda^{s-1} + \cdots + \tilde{Q}_1^{(s)}\lambda + \tilde{Q}_0^{(s)}, \\
\tilde{P}^{(s)}(\lambda) &= \tilde{P}_s^{(s)}\lambda^s + \tilde{P}_{s-1}^{(s)}\lambda^{s-1} + \cdots + \tilde{P}_1^{(s)}\lambda + \tilde{P}_0^{(s)},
\end{aligned}
$$

причем будем требовать, чтобы старший коэффициент $\tilde{Q}_s^{(s)}$ полинома $\tilde{Q}^{(s)}(\lambda)$ был невырожденным.

Алгоритм построен с использованием блочных аппроксимаций Паде. Преимуществом алгоритма по сравнению с алгоритмом Монтгомери является более простая процедура вычисления скалярных произведений V^TAV без участия блоков из пространства Крылова. Это приводит к лучшим параллельным свойствам и, в конце концов, к тому, что время работы сокращается практически до естественной границы - времени на построение пространства Крылова.

Пусть снова требуется решить систему линейных уравнений

$$
Ax = B, A \in \mathbb{F}^{N\times N}, b \in \mathbb{F}^{N\times n}, \tag{6.12}
$$

где A - симметричная матрица.

Для матричного многочлена $R(\lambda) \in \mathbb{F}^{n\times n}[\lambda], n \in \mathbb{N}$ обозначим R_i коэффициент при λ^i, то есть $R(\lambda) = \sum_{i=0}^{degR(\lambda)} R_i\lambda^i$. Пусть также

$$
R(A,B) = \sum_{i=0}^{degR(\lambda)} A^iBR_i,
$$

где $B \in \mathbb{F}^{N\times n}, n \leq N$, а n - длина машинного слова (в этом разделе будем считать, что $n = 64$). Обозначим $W(B) \in \mathbb{F}^{N\times n'}, n' \leq n$ подсовокупность вектор-столбцов матричного блока B такую, что матрица $(W(B))^tAW(B)$ невырождена. Эта подсовокупность может быть построена, например, при

помощи алгоритма из параграфа 8 работы [145]. Пусть, как и выше, O_n, I_n - соответственно нулевая и единичная матрицы из $\mathbb{F}^{n\times n}$.

Рассмотрим случайную матрицу $B \in \mathbb{F}^{N\times n}$. Пусть

$$\alpha = \sum_{i=0}^{\infty} \alpha_i \lambda^{-i}, \alpha_i = B^t A^i B. \tag{6.13}$$

Определим скалярное произведение двух матричных многочленов $(\varphi(\lambda), \psi(\lambda)) \in \mathbb{F}^{n\times n}$ как коэффициент при λ^{-1} в произведении $\varphi(\lambda)^t \alpha \psi(\lambda)$. Обозначим $(C, D) = C^t A D \in \mathbb{F}^{n\times n}$ матрицу попарных A-скалярных произведений вектор-столбцов матриц $C, D \in \mathbb{F}^{N\times n}$. Из этого определения и симметричности матрицы A следует, в частности, что $(C, D) = (D, C)^t$.

Лемма 6.1. *[71, 72]* $(\varphi(A, B), \psi(A, B)) = (\varphi(\lambda), \psi(\lambda))$.

Доказательство. Согласно введённых определений правая и левая часть линейны по φ и ψ, поэтому достаточно доказать лемму при $\varphi = \lambda^i, \psi = \lambda^j, i, j \in \mathbb{N} \cup \{0\}$. В этом случае правая часть равна $B^t A^{i+j+1} B$, а левая

$$(A^i B)^t A A^j B,$$

что, ввиду симметричности матрицы A, одно и тоже. Лемма доказана.

Пусть имеются следующие матричные приближения Паде

$$\begin{gathered} \alpha(\lambda) Q^{(t-1)}(\lambda) - P^{(t-1)}(\lambda) = \sum_{i=t}^{\infty} \tau_i \lambda^{-i}, \\ \alpha(\lambda) Q^{(t)}(\lambda) - P^{(t)}(\lambda) = \sum_{i=t+1}^{\infty} \rho_i \lambda^{-i}, \\ deg Q^{(s)}(\lambda) \le s, deg P^{(s)}(\lambda) \le s, s \in \{t-1, t\}. \end{gathered} \tag{6.14}$$

Здесь λ - переменная, буквы τ_i, ρ_i обозначают матрицы из $\mathbb{F}^{n\times n}$, а $Q^{(s)}(\lambda), P^{(s)}(\lambda) \in \mathbb{F}^{n\times n}[\lambda]$ - матричные многочлены. $Q^{(-1)} = O_n, P^{(-1)} = I_n, Q^{(0)} = I_n, P^{(0)} = \alpha_0$.

Рассмотрим задачу построения следующего $(t+1)$-ого приближения с аналогичными оценками на степень и порядок. По аналогии с одномерным случаем рассмотрим матричный многочлен первой степени

$$b_{t+1}(\lambda) = \nu_1\lambda + \nu_0, \nu_i \in \mathbb{F}^{n\times n}.$$

Домножим второе из матричных уравнений (6.14) на этот многочлен справа и сложим с первым. Обозначим

$$\begin{aligned} Q^{(t+1)}(\lambda) &= Q^{(t)}(\lambda)b_{t+1}(\lambda) + Q^{(t-1)}(\lambda), \\ P^{(t+1)}(\lambda) &= P^{(t)}(\lambda)b_{t+1}(\lambda) + P^{(t-1)}(\lambda). \end{aligned} \tag{6.15}$$

Попытаемся обратить в ноль первые коэффициенты получившегося справа ряда.

$$\alpha(\lambda)Q^{(t+1)}(\lambda) - P^{(t+1)}(\lambda) = \sum_{i=t+2}^{\infty} \gamma_i\lambda^{-i}. \tag{6.16}$$

Тогда получим следующую систему линейных матричных уравнений:

$$\begin{cases} \tau_t = & -\rho_{t+1}\nu_1 \\ \tau_{t+1} = & -\rho_{t+1}\nu_0 - \rho_{t+2}\nu_1. \end{cases} \tag{6.17}$$

Решив эту систему, получим следующее выражение для первого ненулевого коэффициента $(t+1)$-го матричного приближения.

$$\gamma_{t+2} = \tau_{t+2} + \rho_{t+2}\nu_0 + \rho_{t+3}\nu_1. \tag{6.18}$$

Для обеспечения разрешимости относительно ν_1, ν_0 системы (6.17) в случае, когда $rang\rho_{t+1} < n$ нужно применять дополнительные вычисления, чтобы сделать шаг алгоритма. Данная ситуация означает, что среди приближений порядка t есть приближения большего порядка.

Приближения большего порядка, в принципе, можно было бы использовать в нескольких следующих шагах. К сожалению, повторное использование

хороших приближений в следующем шаге приводит к появлению чисто нулевых приближений, после чего алгоритм быстро вырождается.

Причину, а также способ устранения этого явления помогает найти следующая конструкция.

Лемма 6.2. *[72] Ранг по столбцам над полем матричного полинома* $Q_t^{(t)}(\lambda)$ *равен рангу следующей матрицы*

$$\begin{pmatrix} Q_t^{(t)} \\ Q_{t-1}^{(t)} \\ . \\ . \\ . \\ Q_0^{(t)} \end{pmatrix} \tag{6.19}$$

Доказательство. Элементарным преобразованиям столбцов матричного полинома $Q^{(t)}(\lambda)$ соответствуют те же преобразования столбцов матрицы (6.19). Приведём матрицу старшего коэффициента $Q_t^{(t)}$ к виду, в котором первые слева столбцы линейно независимы, а остальные нулевые. Столбцы матрицы $Q_{t-1}^{(t)}$, стоящие под нулевыми столбцами $Q_t^{(t)}$, также приведём к виду, в котором слева стоят линейно независимые, а справа нулевые, и т.д.. Из того, что столбцы разной степени линейно независимы над полем коэффициентов следует, что коранг обоих матриц в условиях леммы равен числу чисто нулевых столбцов, получившихся после окончания описанной процедуры. Лемма доказана.

Систему уравнений (6.17) и преобразование (6.15) запишем в следующем матричном виде

$$\begin{pmatrix} \rho_{t+1} & O_n & \tau_t \\ \rho_{t+2} & \rho_{t+1} & \tau_{t+1} \\ Q_t^{(t)} & O_n & O_n \\ Q_{t-1}^{(t)} & Q_t^{(t)} & O_n \\ Q_{t-2}^{(t)} & Q_{t-1}^{(t)} & Q_{t-1}^{(t-1)} \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \end{pmatrix} \begin{pmatrix} \nu_1 \\ \nu_0 \\ I_n \end{pmatrix} = \begin{pmatrix} O_n \\ O_n \\ Q_{t+1}^{(t+1)} \\ Q_t^{(t+1)} \\ Q_{t-1}^{(t+1)} \\ \cdot \\ \cdot \\ \cdot \end{pmatrix} \tag{6.20}$$

Верхние блочные строки левой матрицы в (6.20), содержащие коэффициенты рядов, будем называть верхней частью, а остальные, содержащие коэффициенты приближений, - нижней частью. Если предположить, что приближение с номером t имеет по столбцам нетривиальный коранг $l > 0$, а приближение с номером $t-1$ - коранг $l' \geq 0$, то после некоторого невырожденного преобразования столбцов вся матрица нижней части будет иметь $2l + l'$ нулевых столбцов. Верхняя часть, ввиду того, что число строк на $n = 64$ меньше числа столбцов, как правило, имеет полный строчный ранг. Если привести её к треугольному виду, то при помощи Гауссовых исключений можно в нижней части сделать нулевыми ещё $2 \cdot 64$ столбцов, стоящих под лидирующими элементами верхней части. Ранг преобразованной таким образом нижней части, а значит, и ранг результата будет не больше $64 - (2l + l')$.

Рассмотрим аналогичные (6.20) преобразования с большим количеством $(k+3)$ блочных столбцов при $t \geq k \geq 0$:

$$\begin{pmatrix} O_n & O_n & O_n & \dots & \sigma_{t-k} \\ \dots & \dots & \dots & \dots & \dots \\ O_n & O_n & O_n & \dots & \sigma_{t-1} \\ \rho_{t+1} & O_n & \tau_t & \dots & \sigma_t \\ \rho_{t+2} & \rho_{t+1} & \tau_{t+1} & \dots & \sigma_{t+1} \\ Q_t^{(t)} & O_n & O_n & \dots & O_n \\ Q_{t-1}^{(t)} & Q_t^{(t)} & O_n & \dots & O_n \\ Q_{t-2}^{(t)} & Q_{t-1}^{(t)} & Q_{t-1}^{(t-1)} & \dots & O_n \\ \dots & \dots & \dots & \dots & \dots \\ Q_{t-k-2}^{(t)} & Q_{t-k-1}^{(t)} & Q_{t-k-1}^{(t-1)} & \dots & Q_{t-k-1}^{(t-k-1)} \\ . & . & . & \dots & . \\ . & . & . & \dots & . \\ . & . & . & \dots & . \end{pmatrix} \begin{pmatrix} \nu_1 \\ \nu_0 \\ \nu_{-1} \\ . \\ . \\ . \\ \nu_{-k-1} \end{pmatrix} = \begin{pmatrix} O_n \\ . \\ . \\ . \\ O_n \\ Q_{t+1}^{(t+1)} \\ . \\ . \\ . \\ Q_{t-k-1}^{(t+1)} \\ . \\ . \\ . \end{pmatrix} \tag{6.21}$$

Ввиду сильно выраженной прямоугольной формы верхней части, она, как правило, имеет полный строчный ранг равный $(k+2)\cdot 64$, в то время как ранг нижней части не превосходит $(k+3)\cdot 64$. Таким образом, для получения полного ранга в столбце справа достаточно, чтобы разность между рангами верхней и нижней части равнялась 64. Для этого достаточно добиться того, чтобы нижняя часть левой матрицы имела полный ранг, например, чтобы диагональные блоки, кроме последнего, были невырождены, а приближение $Q^{(t-k-1)}(\lambda)$ имело ранг n над $\mathbb{F}$. С этой целью можно применить преобразования, не меняющие структуру рассматриваемой матрицы: элементарные преобразования с отдельными приближениями i-го шага, приводящими верхний блок к виду, в котором первые слева столбцы линейно независимы, а остальные нулевые, а затем умножение отдельных приближений i-го шага, имеющих меньшую степень, на λ. Если за одну итерацию полного ранга достичь не удалось, можно повторить эту процедуру, но не более $i-t+k+1$ раз. При $i=t$,

также как и при $i = t-1$, не более k раз. В результате умножения на λ содержимое соответствующего столбца в нижней и верхней части сдвигается на одну блочную позицию вверх и необходимо, чтобы ненулевые элементы не вышли за границы рассматриваемой матрицы. Коэффициенты преобразованных приближений вместе с одним непреобразованным могут быть скомпонованы в систему, аналогичную системе (6.21) следующего вида:

$$\begin{pmatrix} * & * & * & \dots & * \\ \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ * & * & * & \dots & * \\ \tilde{Q}_t^{(t)} & O_n & O_n & \dots & O_n \\ \tilde{Q}_{t-1}^{(t)} & \tilde{Q}_t^{(t)} & O_n & \dots & O_n \\ \tilde{Q}_{t-2}^{(t)} & \tilde{Q}_{t-1}^{(t)} & \tilde{Q}_{t-1}^{(t-1)} & \dots & O_n \\ \dots & \dots & \dots & \dots & \dots \\ \tilde{Q}_{t-k-2}^{(t)} & \tilde{Q}_{t-k-1}^{(t)} & \tilde{Q}_{t-k-1}^{(t-1)} & \dots & Q_{t-k-1}^{(t-k-1)} \\ . & . & . & \dots & . \\ . & . & . & \dots & . \\ . & . & . & \dots & . \end{pmatrix} \begin{pmatrix} \nu_1 \\ \nu_0 \\ \nu_{-1} \\ . \\ . \\ . \\ \nu_{-k-1} \end{pmatrix} = \begin{pmatrix} O_n \\ . \\ . \\ . \\ O_n \\ Q_{t+1}^{(t+1)} \\ . \\ . \\ . \\ Q_{t-k-1}^{(t+1)} \\ . \\ . \\ . \end{pmatrix} \tag{6.22}$$

Решения этой системы всегда дают новое приближение $Q^{(t+1)}(\lambda)$ полного ранга над $\mathbb{F}$. Преобразуя его преобразованиями столбцов и умножениями некоторых из них на λ, получаем приближение $\tilde{Q}^{(t+1)}(\lambda)$ с невырожденным старшим коэффициентом. Последующая доортогонализация строит из этих приближений ортогональный базис пространства Крылова (см. далее).

Численные эксперименты, проведённые моим учеником И.А.Поповяном, показывают, что указанные преобразования достаточно эффективны. Алгоритм, позволяющий наращивать число столбцов до $k = 9$, успешно обрабатывает

квадратную симметричную матрицу размера $N = 2000$ с вероятностью около 75 процентов, при этом с увеличением размера матрицы при сохранении значения вероятности максимальное число используемых столбцов растёт логарифмически. Кроме того, среднее число столбцов по всем шагам оказывается около 4.

Очевидно, что любое приближение с номером t может быть получено как линейная комбинация $\lambda^i Q^{(0)}, i = 0, 1, \ldots, t$ с коэффициентами $\upsilon_i \in \mathbb{F}^{n\times n}$, определёнными системой:

$$\begin{pmatrix} \alpha_{t+1} & \ldots & \alpha_2 & \alpha_1 \\ \alpha_{t+2} & \ldots & \alpha_3 & \alpha_2 \\ \ldots & \ldots & \ldots & \ldots \\ \alpha_{2t} & \ldots & \alpha_{t+1} & \alpha_t \\ I_n & \ldots & O_n & O_n \\ \ldots & \ldots & \ldots & \ldots \\ O_n & \ldots & I_n & O_n \\ O_n & \ldots & O_n & I_n \end{pmatrix} \begin{pmatrix} \upsilon_t \\ . \\ . \\ . \\ \upsilon_1 \\ \upsilon_0 \end{pmatrix} = \begin{pmatrix} O_n \\ . \\ . \\ O_n \\ Q_{t+1}^{(t)} \\ . \\ . \\ Q_0^{(t)} \end{pmatrix} \tag{6.23}$$

Верхняя часть этой матрицы имеет размер t на $t+1$ блок, поэтому она имеет блочное решение с полным рангом, а получившееся матричное приближение будет иметь полный ранг над полем.

Ввиду невырожденности старших коэффициентов многочленов $\tilde{Q}^{(i)}(\lambda)$, выполнено

$$\langle \tilde{Q}^{(i)}(A,B), i = 0, 1, ..., t\rangle = \langle I^{(i)}(A,B), i = 0, 1, ..., t\rangle,$$

где $I^{(i)}(\lambda) = I_n\lambda^i$, которое по определению савпадает с $\langle A^iB, i = 0, 1, ..., t\rangle$ и при достаточно больших t совпадает со всем пространством Крылова.

Лемма 6.3. *[72] Пусть при $k, m \in \mathbb{N}$ имеется прямоугольная неоднородная система вида*

$$A'X = B'; X \in \mathbb{F}^{(k+1)m\times m},, B \in \mathbb{F}^{km\times m}$$

со случайной матрицей $A' \in \mathbb{F}^{km\times(k+1)m}$.

Вероятность её разрешимости относительно X *не меньше, чем* $1 - \frac{1}{2^m}$.

Действительно, в случае, когда матрица, стоящая слева в рассматриваемом матричном равенстве, имеет полный строчный ранг, эта система разрешима. Таким образом, достаточно оценить вероятность $T(km)$ того, что матрица A' имеет полный строчный ранг. Докажем это по индукции для подматриц первых строк матрицы A'. Первые $t+1$ строк являются совокупностью линейно независимых строк тогда и только тогда, когда первые t из них линейно независимы, а последняя не лежит в их линейной оболочке. Поэтому рассматриваемая вероятность $T(t)$, очевидно, удовлетворяет следующему рекуррентному уравнению

$$T(t+1) = T(t)(1 - \frac{1}{2^{(k+1)m-t}}).$$

Поэтому

$$T(km) = (1 - \frac{1}{2^{m+1}}) \dots (1 - \frac{1}{2^{(k+1)m}}).$$

Поскольку при $A, B > 0$ выполнено $(1-A)(1-B) \geq (1-(A+B))$, то из последнего равенства следует, что

$$T(km) \geq 1 - \frac{1}{2^m}.$$

Лемма доказана.

Из этой леммы следует, что системы (6.21), (6.22) и (6.23) эффективно решаются.

Согласно результатам, приведённым в [145] для вероятностей P_k того, что случайная симметричная матрица из $\mathbb{F}^{n\times n}$ имеет ранг $k, k \leq n = 64$, имеются

следующие значения:

$$P_k = \left[(1-\tfrac{1}{2})(1-\tfrac{1}{2^3})(1-\tfrac{1}{2^5})\dots\right] \tfrac{1}{(2-1)(2^2-1)(2^3-1)\dots(2^{n-k}-1)} < \\ < \tfrac{1}{2\cdot 2^2\cdot 2^3\cdot\dots\cdot 2^{n-k-1}} = 2^{-\frac{(n-k)(n-k-1)}{2}}, \qquad (6.24)$$

$$P_{64} \approx \tfrac{42}{100}, P_{63} \approx \tfrac{42}{100}, P_{62} \approx \tfrac{13}{100}, P_{61} \approx \tfrac{2}{100}, \textstyle\sum_{k=0}^{60} P_k < \tfrac{1}{100}.$$

Для несимметричных матриц (см. например [151]) ситуация примерно такая же

$$P'_{64} \approx \tfrac{28{,}8}{100}, P'_{63} \approx \tfrac{51}{100}. \qquad (6.25)$$

Эти оценки показывают, что средние и максимальные значения $\delta(i)$ и числа столбцов в матрицах на каждом шаге невелики и не оказывают существенного влияния на сложность метода.

Заметим, что матрицы в левой части (6.21) и (6.22) отличаются от матрицы в левой части (6.23) при $k = t-1$, лишь умножением справа на некоторую блочнонижнетреугольную квадратную матрицу. Если при помощи системы (6.23) получить приближения с номерами $1, 2, \dots, t-1$, то в дальнейшем для построения следующих приближений можно использовать преобразования шага с числом приближений от 2 до $t+1$.

Таким образом, в случае, если на каком-то шаге придётся использовать все предыдущие приближения, можно вместо системы (6.22) решать систему (6.23). Тем самым, данный алгоритм, в отличие от алгоритма статьи [145], срабатывает всегда, правда, может потребовать большего времени и памяти.

Повторяем указанные процедуры, как и в алгоритме Монтгомери [145], до полного вырождения последовательности матриц $W(\tilde{Q}^{(i)}(A,B))$. По построению

$$\langle \tilde{Q}^{(i)}(A,B)\rangle_{i=0,\dots,k} = \langle A^i B\rangle_{i=0,\dots,k}.$$

Поэтому

$$Q^{(k+1)}(A,B) \in \langle \tilde{Q}^{(i)}(A,B)\rangle_{i=0,\ldots,k}$$

тогда и только тогда, когда

$$A^{k+1}B \in \langle A^i B\rangle_{i=0,\ldots,k}.$$

В случае, если алгоритм с использованием ограниченного числа предыдущих приближений на каком-то шаге не срабатывает, и если при этом решение системы (6.12) не лежит в линейной оболочке уже построенных векторов, можно начать с начала с другой матрицей B, или рандомизировать матрицу A, применяя соображения описанные в [153, 154]. Отметим, однако, что несрабатывание алгоритма возможно только в случае, если нам не удалось описанными выше процедурами добиться того, чтобы диагональные блоки в левой части матрицы (6.21) были невырожденными. Если считать коэффициенты получающихся приближений Паде случайными, вероятность этого события очевидно экспоненциально убывает с ростом числа используемых приближений.

Заметим, что при любых $j,t : 0 \le j \le t \in \{0,1,\ldots,\frac{N}{n}-1\}$

$$(Q^{(j)}(A,B), Q^{(t+1)}(A,B)) = O_n,$$

так как коэффициент при λ^{-1} в ряде

$$(Q^{(j)}(\lambda))^T \alpha(\lambda) Q^{(t+1)}(\lambda) = (Q^{(j)}(\lambda))^T (\sum_{i=t+2}^{\infty} \gamma_i \lambda^{-i}) \tag{6.26}$$

равен нулю. Поэтому столбцы матриц $Q^{(i)}(A,B)$ лежат в попарно A-ортогональных подпространствах, а их количество, или, точнее сказать, количество среди них таких, которые дают невырожденную матрицу попарных A-скалярных произведений, должно быть приблизительно равно размерности всего пространства. Для того, чтобы этого добиться, описанные процедуры нужно повторять несколько более чем $\frac{N}{n}$ раз. Обозначим $V_t = W(\tilde{Q}^{(t)}(A,B))$.

Ожидаемый размер (ранг) матрицы (V_t, V_t) предположительно будет лишь чуть меньше n. Для построения всего пространства Крылова необходимо использовать все столбцы $\tilde{Q}(t)(A,B)$. Более подробно этот процесс будет описан ниже под названием "дополнительная ортогонализация".

Если он в среднем оказался, как для случайных матриц, меньшим приблизительно на единицу, то при решении невырожденной системы линейных уравнений (6.12) можно считать достаточным количество указанных итераций, приблизительно равное $\frac{N}{n}+\frac{N}{n^2}+\cdots=\frac{N}{n-1}$. В случае вырожденной системы следует заменить N на $rangA$.

Решение системы (6.12) можно получить по теореме 6.4.

В ситуации, когда связь быстрая, выгоднее уменьшить диапазон вычисляемых A^iB вдвое до $i=1,\ldots,\frac{N}{n}$ и использовать для построения коэффициентов ряда формулу $\alpha_{i+j}=(A^iB)^TA^jB=B^TA^{i+j}B, j=i,i+1$. Это даёт вдвое меньшее число умножений матрицы на блок, то есть равное примерно $\frac{N}{n}$, как в алгоритме Монтгомери.

Поскольку

$$Q^{(t+1)}(\lambda)=\lambda\tilde{Q}^{(t)}(\lambda)\nu_1+\tilde{Q}^{(t)}(\lambda)\nu_0+$$
$$+\tilde{Q}^{(t-1)}(\lambda)\nu_{-1}+\cdots+Q^{(t-k-1)}(\lambda)\nu_{-k-1},$$

имеем

$$Q^{(t+1)}(A,B)=A\tilde{Q}^{(t)}(A,B)\tilde{\nu}_1+\tilde{Q}^{(t)}(A,B)\tilde{\nu}_0+$$
$$+\tilde{Q}^{(t-1)}(A,B)\tilde{\nu}_{-1}+\cdots+Q^{(t-k-1)}(A,B)\tilde{\nu}_{-k-1}. \qquad (6.27)$$

Вычисление по этим формулам требует примерно $\frac{N}{n}$ умножений на матрицу A. Однако, поскольку для построения первых приближений требуются только первые коэффициенты ряда $\alpha(\lambda)$. Процесс вычисления этих коэффициентов можно вести практически параллельно, и время на дополнительные умножения на матрицу A в этом случае можно не учитывать. Для того, чтобы получить

$\tilde{Q}^{(i)}(A,B)$, в каждом шаге потребуется ещё несколько умножений на матрицу A. Далее вся эта процедура будет существенно оптимизирована.

Другой вариант, это сгруппировать повторный ряд умножений на матрицу A, как в алгоритме Видемана-Копперсмита в конце алгоритма. Для этого в последнюю формулу вместо блоков $\tilde{Q}^{(i)}(A,B)$ следует подставлять вектора их координат в базисе A^jB в диапазоне примерно $j = 1,\ldots,\frac{N}{n}$, а вместо $A\tilde{Q}^{(i)}(A,B)$ - соответствующий i-й сдвинутый вектор координат. После подстановки координат этих блоков в формулу для решений получим координаты решения в том же базисе. Так как степень многочленов $\tilde{Q}^{(i)}(A,B)$ равна i, то есть совпадает с их индексом, дополнительных умножений на матрицу A, о которых шла речь в предыдущем абзаце, здесь не потребуется, но затем надо будет повторно вычислить базис $A^iB, i = 0,1,\ldots.\frac{N}{n} + \varepsilon$ и, последовательно суммируя, получить решение. Вместо повторного вычисления можно запомнить эти блоки после их первичного вычисления, однако это потребует большого объёма памяти. Об этом варианте мы поговорим в главе, посвящённой новому алгоритму типа Видемана-Копперсмита. В дальнейшем будет предложен способ экономии оперативной памяти, "синхронизация".

Итак, в первом варианте количество необходимых умножений на матрицу A, влияющих на время, можно считать примерно равным $(1+\theta)\frac{N}{n}$, где θ - число дополнительных умножений на A при построении очередного блока векторов пространства Крылова. Для того, чтобы доказать оценку на количество операций, необходимых для всех умножений матрицы A на блок в виде $O(\frac{dN^2}{n})$, достаточно применить очевидную оценку для количества операций для одного умножения (построчно) в виде $O(dN)$. Численный эксперимент показывает, что среднее число по всем шагам θ примерно равно 1,5.

Ещё примерно $2\frac{N}{n}$ умножений плотных блоков друг на друга потребуется для вычисления всех необходимых $\alpha_i, i = 1,2,\ldots,\frac{2N}{n}$ в равенстве (6.13). Важно отметить, что эти вычисления могут быть проделаны независимо друг

от друга и от основной части алгоритма. Удвоенный набор коэффициентов необходим для получения возможности вычислять необходимые в формулах (6.22) коэффициенты правых частей формул (6.11).

Количество элементов α_i, которые нужно использовать для вычисления нескольких первых элементов в правых частях равенств (6.14) в виде свёрток на шаге t не превосходит $t+1 \leq \frac{N}{n}+\varepsilon$, а количество блочных операций (построчное умножение друг на друга матриц размера $n \times n$) для этих вычислений во всём алгоритме получаем суммированием по всем t в виде $O((\frac{N}{n})^2)$, при сложности одной блочной операции не более $O(n^2)$ (здесь n - длина машинного слова). Итого получаем $O(N^2)$. Для вычислений правых частей по формулам (6.21), а также для вычисления коэффициентов блоков, полученных из приближений и построения решения, требуется $O(\frac{N}{n})$ умножений плотных блоков из $\mathbb{F}^{N\times n}$ на маленькие матрицы из $\mathbb{F}^{n\times n}$. Тривиальная оценка построчного умножения $O(Nn)$ снова даёт $O(N^2)$. При этом, как будет показано ниже, существует алгоритм, реализующий умножение плотного блока размера $N \times n$ на маленькую матрицу из $\mathbb{F}^{n\times n}$ (линейные комбинации) за время $O(\frac{N}{nlogN})$, а также алгоритм, реализующий с такой же сложностью умножение плотных блоков размера $N \times n$ на себя (скалярные произведения). Отметим также, что эти операции хорошо параллелизуются. Сложение плотных блоков осуществляется методом циклической пересылки, осуществляющим параллельное сложение одновременно нескольких векторов [155]. Считая это время за асимптотически меньшее, мы не включаем его в итоговую оценку сложности. Тем самым, окончательная оценка сложности этой первой версии алгоритма имеет главный член вида $(1+\theta)\frac{N^2 d}{n}$.

Для того, чтобы построенное нами пространство совпадало с пространством Крылова, необходимо использовать вектора, соответствующие приближениям $\tilde{Q}^{(i)}(\lambda)$, полученным из $Q^{(i)}(\lambda)$ с использованием невырожденных преобразований столбцов и умножения на λ. Матрица, составленная из коэффициентов $\tilde{Q}^{(i)}(\lambda), i = 0, \ldots, t$:

$$\begin{pmatrix} \tilde{Q}_t^{(t)} & O_n & \dots & O_n \\ \tilde{Q}_{t-1}^{(t)} & \tilde{Q}_{t-1}^{(t-1)} & \dots & O_n \\ \dots & \dots & \dots & \dots \\ \tilde{Q}_0^{(t)} & \tilde{Q}_0^{(t-1)} & \dots & \tilde{Q}_0^{(0)} \end{pmatrix}, \tag{6.28}$$

где $\tilde{Q}_0^{(0)} = Q^{(0)}$, невырождена, ввиду невырожденности диагональных элементов, поэтому, обозначая $+$ — линейное пространство, порождённое слагаемыми, получим:

$$\langle \tilde{Q}^{(m)}(A,B) \rangle + \dots + \langle \tilde{Q}^{(0)}(A,B) \rangle = \langle \lambda^m Q^{(0)}(A,B) \rangle + \dots + \langle Q^{(0)}(A,B) \rangle.$$

Напомним, что угловыми скобками мы обозначаем линейное пространство, образованное столбцами рассматриваемых матриц, а $\tilde{Q}^{(0)} = Q^{(0)}$. Поскольку $(\lambda^t Q^{(0)})(A,B) = A^t B$, при m чуть больше $\frac{N}{n}$ мы получим всё пространство Крылова. Добавленные векторы легко ортогонализовать, так как они могут быть не ортогональны только блокам с номерами $t-k$, где k не больше числа использованных итераций при построении $\tilde{Q}^{(t)}$, и скалярные произведения их с построенными ортогональными блоками вычисляются при помощи уже вычисленных маленьких матриц коэффициентов самого приближения и коэффициентов разложения по формулам вида (6.26) (см. далее формальное описание процедуры доортогонализации). Для построения $W_t, t = 1, \dots, m-1$, используемых в формуле для решения (6.7), нужно использовать не сами Паде аппроксимации $Q^{(t)}$, а ортогонализованные $\tilde{Q}^{(t)}(A,B)$. При этом при использовании алгоритма из параграфа 8 работы [145], если какие-то из столбцов $\tilde{Q}^{(t)}(A,B)$ не удалось включить в W_t и W_{t+1}, это не приводит к остановке алгоритма - их можно включать в W_i для любого $i = t+2, \dots, m-1$.

Доортогонализация.

Теперь опишем процесс дополнительной ортогонализации блоков $V_i = \tilde{Q}^{(i)}(A,B), i = 0,1,\ldots$, то есть построение A-ортогонального базиса из квази A-ортогонального. Предположим, что нам известны $V_i = \tilde{Q}^{(i)}(A,B), i = 0,1,\ldots,t+1$, тогда из равенств (6.10) и леммы 6.1 имеем: ${V_{t+1-j}}^T A V_{t+1} =$

$$\begin{cases} 0 \quad \text{при} \quad j - \delta(t+1) \geq 2 \\ \sum_{k-l=-1} \tilde{Q}_k^{(t+1-j)T} \tilde{\alpha}_l^{(t+1)} = \\ = \tilde{Q}_{t+1-\delta(t+1)-1}^{(t+1-j)T} \tilde{\alpha}_{t+1-\delta(t+1)}^{(t+1)} + \cdots + \tilde{Q}_{t+1-j}^{(t+1-j)T} \tilde{\alpha}_{t+2-j}^{(t+1)} \quad \text{иначе} \end{cases} \tag{6.29}$$

Несмотря на видимую громоздкость этой формулы, она легко вычисляется, поскольку содержит матрицы маленького размера. В случае, если $\delta(i)$ не превосходит некоторой постоянной границы δ, будем хранить 2δ старших (первых) коэффициентов разложений в правых частях (6.11). Тогда по рекуррентным формулам (6.22) можно построить $2\delta - (k+2)$ старших коэффициентов правой части разложения вида (6.10), соответствующего новому $t+1$-ому приближению. Затем производятся описанные выше преобразования столбцов с умножением на λ для вычисленного куска ряда $\alpha^{(t+1)}(\lambda)$ и многочлена $Q^{(t+1)}(\lambda)$. Недостающие $k+1$ коэффициентов вычисляется по формуле (6.11) при помощи свёртки многочлена $\tilde{Q}^{t+1}(\lambda)$ с куском ряда $\alpha(\lambda)$. Для этого число вычисленных коэффициентов ряда должно быть не меньше $\frac{2N}{n} + \delta$ с некоторым небольшим δ.

Завершение алгоритма.

Пусть m таково, что матрица попарных A-скалярных произведений векторов, полученных из $\tilde{Q}^{(m)}(A,B)$, а также не вошедших в $W_i, i = 1,\ldots,m-1$, равна нулю. Линейно независимые из этих векторов поместим в W_m. Проделаем ещё несколько шагов, ортогонализуя получающиеся вектора относительно $W_i, i = 1,\ldots,m-1$. Если получаться вектора не ортогональные всем уже входящим в W_m, то с их участием можно составить блок с невырожденным A-скалярным произведением, то есть увеличить m на единицу, в противном случае, линейно

независимые с вектор-столбцами W_m надо включить в W_m. Построение очередных приближений следует прекратить, если на некотором шаге после ортогонализации относительно $W_i, i = 1, \ldots, m-1$ все полученные вектора-столбцы лежат в линейном пространстве тех, которые уже вошли в $\langle W_m \rangle$. Другими словами умножение на A не приводит к расширению рассматриваемого пространства векторов, а значит мы исчерпали всё пространство Крылова. Таким образом алгоритм всегда закончит свою работу, построив пространство Крылова. Как и в алгоритме Монтгомери, предполагается, что число всех необходимых шагов превосходит $\frac{N}{n}$ незначительно. То есть на каждом шаге алгоритма размерность пространства Крылова подрастает на величину, близкую к n.

Для того, чтобы получить оценку на количество операций, рассмотрим, какие параллельные процессы использует данный алгоритм и выпишем главные члены верхних оценок их сложности:

1. $\frac{N}{n}$ умножений на матрицу A для построения $\alpha_i, i = 1, \ldots, \frac{2N}{n}$ (последовательно со сложностью $C_2 \frac{N^2 d}{n}$, согласно Монтгомери, умножение матрицы A на блок из $\mathbb{F}^{N \times n}$ требует $C_2 N d$ операций). Напомним, что здесь для уменьшения числа умножений на матрицу A применяется формула $\alpha_{i+j} = (A^i B)^t A^j B$.

2. $\frac{2N}{n}$ умножений плотных блоков из $\mathbb{F}^{n \times N}$ друг на друга для построения $\alpha_i, i = 1, \ldots, \frac{2N}{n}$ (параллельно, то есть эти вычисления могут быть проделаны независимо без задержки времени).

3. Примерно $\frac{4{,}85N}{n}$ вычислений коэффициентов разложения при помощи свёрток для построения $\tilde{Q}^{(i)}, i = 1, \ldots, m$ (последовательно со сложностью $\frac{4{,}85}{2} C_1 N^2$, согласно Монтгомери каждая свёртка из t элементов вычисляется со сложностью $C_1 t n^2$). Число $4,85$ здесь получено из результатов численного эксперимента средняя величина параметра $k+2$ в равенстве (6.22) используемых во всех шагах алгоритма.

4. Вычисление пространства Крылова. Для этого необходимо $\frac{N}{n}$ умножений матрицы A на блоки и в среднем 1,5 умножения A на блоки в каждом шаге (последовательно со сложностью $2,5C_2\frac{N}{n}Nd$).

   В оценке шагов 3, 4 применены результаты численного эксперимента, о котором я уже упоминал.

5. Ортогонализация пространства Крылова. Примерно $\frac{N}{n}$ вычислений матриц W_i (последовательно). Доортогонализация производится с использованием скалярных произведений, построенных на 3 шаге, при помощи арифметики с маленькими матрицами.

6. Вычисление решения по формуле (6.7). Содержащееся здесь умножение плотных блоков $W_i^t B$ может быть достаточно быстро вычислено по рекуррентной формуле из уже вычисленных скалярных произведений (последовательно) (Этот вариант описан ниже в оптимизированной версии), или при помощи умножения плотных блоков (параллельно).

Максимум по всем перечисленным параллельным процессам достигается в шаге 3 или 4 и равен $max\{\frac{2,85}{2}C_1N^2, 2,5C_2\frac{N}{n}Nd\}$. Как мы увидим ниже, время на производство 3 шага может быть асимптотически уменьшено. Для оптимизации 4-го шага одно из дополнительных умножений на матрицу A можно проделать в этом же шаге, воспользовавшись вырождением $\tilde{\nu}_1$ в системе (6.22).

Действительно, если $Q_{t+1}^{(t+1)}$- невырождена, то её не надо "поднимать то есть производить дополнительные умножения на λ и A. В любом случае $corang\tilde{\nu}_1 = corang\tilde{Q}_{t+1}^{(t+1)}$, и одно "поднятие"мы можем осуществить вместе с умножением A на $\tilde{Q}_t^{(t)}(A,B)\tilde{\nu}_1$. Далее будет изложена версия рассматриваемого алгоритма, в которой процедура "поднятия"отсутствует, дополнительных умножений на матрицу A нет, то есть можно положить в оценках $\theta = 0$.

Важным достоинством данного алгоритма по сравнению с алгоритмом Монтгомери является более простая процедура вычисления скалярных

произведений. Это вычисление сводится к подсчёту нескольких слагаемых в остатке разложения, которое даёт соответствующее приближение Паде, а именно тех, которые участвуют в формировании коэффициента при λ^{-1} в правой части формулы (6.29). Сами эти слагаемые вычисляются, как было указано выше, в виде свёртки, то есть фактически тоже путём умножения плотных блоков. Однако эти блоки не постоянной большой длины, как в алгоритме Монтгомери, а меньшей, пропорциональной номеру шага. Далее будет показано как эту длину ещё уменьшить.

Вместо блоков, образующих ортогональный базис пространства Крылова можно вычислять, как мы уже обсуждали, только их коэффициенты в базисе A^iB, с помощью которых скалярные произведения этих блоков могут быть выражены через скалярные произведения элементов базиса, то есть через α_i. В описанной далее оптимизированной версии в процессе работы алгоритма периодически производится процедура синхронизации в результате которой исходный базис заменяется на базис ортогонального дополнения к уже построенной части пространства Крылова. Далее этот базис ортогонализуется, причем новый ортогональный базис выражается через старый при помощи небольшого числа коэффициентов, также как в начале алгоритма.

Отметим также, что для успешной работы алгоритма Монтгомери необходимо, чтобы для матриц S_i, построенных вспомогательным алгоритмом, выполнялось условие (2.7) (см. Приложение), которое приводит к утверждению леммы 2.1 (см. Приложение). Оценок вероятности выполнения этих условий нет. Правда для того, чтобы обойти это место, в литературе предлагается так называемая "look a head"[156] стратегия, то есть использование большего числа предыдущих, уже построенных блоков. Дополнительная информация, получаемая из коэффициентов Паде аппроксимаций и старших коэффициентов соответствующих им разложений, помогает провести в этом месте более точный анализ, получить теоретические оценки вероятности срабатывания

всего алгоритма (см. ниже), а в дальнейшем совсем отказаться от проблем со срабатыванием шага алгоритма в версии [74] (см. следующий подраздел, посвящённый оптимизированной версии).

Вспомогательный алгоритм (доказательство корректности).

Рассмотрим симметричную матрицу над полем из двух элементов $A \in \mathbb{F}^{N\times N}$ и блок $V \in \mathbb{F}^{N\times n}$, где n - длина машинного слова. Обозначим $T = V^T AV$. В статье [145] предложен алгоритм, который по T и некоторому подмножеству $\overline{S}_1 = \{\mathbf{s}_{11}, \ldots, \mathbf{s}_{1m}\} \subseteq \{1, \ldots, n\}, m \leq n$ выдаёт набор $\overline{S}_2 = \{\mathbf{s}_{21}, \ldots, \mathbf{s}_{2k}\} \subseteq \{1, \ldots, n\}, k \leq n$ и матрицу $W^{inv} = S_2(S_2{}^T T S_2)^{-1} S_2{}^T$, где $S_2 = (\beta_{ij}) \in \mathbb{F}^{n\times k}, \beta_{ij} = 1 \Leftrightarrow j = \mathbf{s}_{2i}$, выполнено условие (2.7) (см. Приложение), и матрица $S_2{}^T T S_2$ – невырождена и имеет максимальный размер при данных условиях. Алгоритм в статье [145] приведён в виде схемы без обоснования. Теоретически эту задачу можно выполнить проще, однако, поскольку данный алгоритм работает достаточно быстро, мы дадим здесь его полное описание и обоснование, сохраняя те допущения, которые сделаны в [145].

Нетрудно понять, что если обозначить также $S_1 = (\alpha_{ij}) \in \mathbb{F}^{n\times m}$, где $\alpha_{ij} = 1 \Leftrightarrow j = \mathbf{s}_{1i}$, то условие (2.7) (см. Приложение) эквивалентно тому, что $\{1, \ldots, n\} \backslash \overline{S}_2 \subseteq \overline{S}_1$, или, что то же самое, $\{1, \ldots, n\} \setminus \overline{S}_1 \subseteq \overline{S}_2$.

В начале алгоритм фактически производит перестановку столбцов (и строк для сохранения симметричности) матрицы T, выставляя в крайнее левое положение столбцы, номера которых не содержатся в множестве $\overline{S}_1$. Поскольку в реальности перестановка заменяется на перенумерацию столбцов, полученную после указанной перестановки, матрицу будем также обозначать T. Дальнейшая работа состоит в последовательных действиях со строками некоторой матрицы из $\mathbb{F}^{N\times 2n}$. Эти действия являются либо элементарными преобразованиями, либо обнулением строки целиком. В начале алгоритма указанная матрица имеет вид

$$M = \left(\underbrace{T}_{M_1} \| \underbrace{I_n}_{M_2} \right).$$

Так как преобразования производятся только со строками, то в течение всего алгоритма правая и левая части матрицы M связаны равенством

$$M_1 = M_2 T. \tag{6.30}$$

Теорема 6.5. *[158] Для произвольной симметричной матрицы* $T \in \mathbb{F}^{n\times n}$ *существует матрица* $S \in \mathbb{F}^{n\times n'}, n' \leq n$, *такая, что* $S^T T S$ *обратима, и* $S^T S = I_{n'}$.

Доказательством служит следующий алгоритм. Алгоритм состоит из трёх циклов. Первый цикл слева направо по столбцам матрицы M_1. Она приводится к верхнетреугольному виду. Обозначим номер текущего столбца i. Второй цикл встроен в первый, идёт по строкам сверху вниз, начиная с i-ой строки. В нём мы ищем строку, имеющую в i-м столбце единицу. Если такая строка найдена, она переставляется с i-й и вычитается из других строк матрицы, имеющих единицу в i-м столбце. Номер i запоминается тем, что в матрицу S записывается i-й столбец единичной матрицы. При этом на i-м месте в матрице M_1 также оказывается i-й столбец единичной матрицы.

Пусть такой строки найти не удалось. Тогда i-й столбец единичной матрицы в S не запоминается, и включается третий цикл, делающий те же преобразования, что и второй, но с $(n+i)$-м столбцом, то есть с i-м столбцом матрицы M_2, при этом делается **допущение**, что в этом столбце найдётся ненулевой элемент (единица), стоящий не выше i-й строки. После перестановки соответствующей строки с i-й и вычитания её из остальных строк, имеющих на $(n+i)$-м месте единицу, вся i-я строка обнуляется.

По окончании работы алгоритма выполняются следующие равенства:

$$M_1 S = S, M_2 = M_2 S S^T, M_2 = S S^T M_2. \tag{6.31}$$

Последние два равенства выполнены потому, что строки и столбцы матрицы M_2, соответствующие индексам, которые не запоминались, наш алгоритм обнуляет. В тоже время, как было указано выше, столбцы матрицы M_1 с индексами, которые запоминались, являются соответствующими столбцами единичной матрицы, поэтому выполняется и первое из этих равенств.

Из равенства (6.30) и первого из равенств (6.31) находим:

$$M_2 TS = S.$$

Подставляя сюда второе равенство из (6.31), получаем:

$$M_2 S S^T T S = S. \tag{6.32}$$

Поскольку в столбцах матрицы S ровно по одному ненулевому элементу (единице), то $S^T S = I_{n'}$, откуда, домножив последнее равенство слева на S^T, получим, что матрица $S^T T S$ обратима.

Теорема доказана.

Следствие 6.1. *[158] По окончании работы алгоритма из предыдущей теоремы имеем*

$$M_2 = W^{inv} = S(S^T T S)^{-1} S^T.$$

Домножив равенство (6.32) справа на $(S^T T S)^{-1}$, получим:

$$M_2 S = S(S^T T S)^{-1}.$$

Если теперь домножить это равенство справа на S^T и воспользоваться снова вторым равенством из (6.31), то окончательно получим искомое равенство. Следствие доказано.

Построенные матрицы S и W^{inv} позволяют строить A-ортогональные проекции на подпространство, порождённое теми столбцами матрицы V, из них

состоит матрица $W = VS$, для которых матрица их попарных A-скалярных произведений $(VS)^T A(VS)$ невырождена. По доказанному

$$W(W^T AW)^{-1} W^T AU = VW^{inv} V^T AU,$$

поэтому, ввиду того, что $B = AX$, равенство (6.7) можно переписать в виде

$$X = \sum_{i=0}^{m-1} V_i W_i^{inv} V_i^T B.$$

Вероятностный анализ

В этом подразделе предлагаются [73] некоторые оценки среднего числа используемых приближений в предложенной выше версии алгоритма построения матричных приближенеий Паде над $\mathbb{F} = GF(2)$.

Матричные многочлены $Q^{(s)}(\lambda) \in \mathbb{F}^{n\times n}[\lambda]$, удовлетворяющие

$$\begin{gathered} \alpha(\lambda)Q^{(s)}(\lambda) - P^{(s)}(\lambda) = \sum_{i=s+1}^{\infty} \rho_i^{(s)} \lambda^{-i}, \\ deg Q^{(s)}(\lambda) \le s, deg P^{(s)}(\lambda) \le s \end{gathered} \tag{6.33}$$

для некоторых $P^{(s)}(\lambda) \in \mathbb{F}^{n\times n}[\lambda]$ будем, как и раньше, называть матричными приближениями Паде с номером s, или просто s приближениями ряда $\alpha(\lambda)$. Обозначим коэффициенты этих приближений в соответствии со следующей формулой.

$$Q^{(s)}(\lambda) = \sum_{i=0}^{s} Q_i^{(s)} \lambda^i, Q_i^{(s)} \in \mathbb{F}^{n\times n}.$$

Пусть $Q_s^{(s)}$ - вырожденная матрица. Рассмотрим ещё раз следующее преобразование, описанное ранее. При помощи элементарных преобразований столбцов матричного полинома $Q^{(s)}(\lambda)$ приведём его старший коэффициент

к виду, где левые столбцы линейно независимы, а остальные равны нулю. Затем умножим столбцы матричного многочлена $Q^{(s)}(\lambda)$, соответствующие нулевым столбцам главного члена, на λ. Будем повторять указанные процедуры до получения невырожденного старшего коэффициента. Обозначим число необходимых итераций ζ_s. Будем считать, что s достаточно велико, чтобы ζ_s было конечным. Предположим, что матричные коэффициенты Паде аппроксимаций случайны и статистически независимы, откуда величины $\zeta_s, s \in \mathbb{N}$ - независимые случайные величины.

Рассмотрим некоторые хорошо известные [151] вероятностные оценки для случайных квадратных матриц над $\mathbb{F}$ фиксированного размера:

$$\rho = P(corankM \geq 2) \leq \frac{1}{4},$$
$$\sigma = P(corankM > 0) \leq 0,72,$$
$$\delta = P(corankM = 0) \leq \frac{1}{2},$$
$$P(corankM = 1) \leq 0,6.$$

Если размер матриц больше, чем 64, имеем $\delta = P(corankM = 0) \in (0,288; 0,29)$. Поскольку с ростом числа итераций число нулевых столбцов не возрастает, имеем

$$P(\zeta_s \leq k) = 1 - P(\zeta_s > k) \geq 1 - \sum_{l=0}^{k} \rho^l \frac{0,6}{2^{k-l}} \geq 1 - \sum_{l=0}^{k} \frac{0,6}{2^{k+l}} \geq 1 - 2\frac{0,6}{2^k} = 1 - \frac{1,2}{2^k}. \tag{6.34}$$

Поскольку

$$P(\zeta_s > 1) \leq P(corankM = 1)\frac{1}{2} + P(corankM \geq 2)0,72 \leq 0,6\frac{1}{2} + \frac{1}{4}0,72 = 0,48,$$

имеем

$$ln(P(\zeta_s \leq 1)) = ln(1 - P(\zeta_s > 1)) \geq ln(1 - 0,48) \geq -0,654. \tag{6.35}$$

Как было показано выше, при $r > t-1 > 0$ выполнение неравенств

$$\zeta_r \le t-1, \zeta_{r-1} \le t-1, \zeta_{r-2} \le t-2, \ldots, \zeta_{r-t+1} \le 1 (\zeta_r = 0, \text{ при } t=1) \qquad (6.36)$$

достаточно для построения аппроксимации с номером $r+1$ при помощи аппроксимаций с номерами $r, r-1, \ldots, r-t$, то есть $t+1$ приближение. Для упрощения переобозначим $\xi_1 = \zeta_r, \xi_2 = \zeta_{r-1}, \ldots, \xi_t = \zeta_{r-t+1}$. Пусть

$$\theta = min\{t \ge 1 : \xi_1 \le t-1, \xi_k \le t-k+1, k = 2, \ldots, t \text{ при } t > 1, \xi_1 = 0 \text{ при } t = 1\},$$

$$\tau = \min\{t \ge 1 : \xi_k \le t-k+1, k = 1, \ldots, t\}.$$

Матожидание $M(\theta)$ — это среднее значение $k+1$ в равенствах (6.22)

Идея доказательства следующей теоремы была предложена А.М.Зубковым.

Теорема 6.6. *[73] В предположении, что матричные коэффициенты Паде аппроксимаций случайны и статистически независимы*

$$M(\theta) \le 7,233.$$

Доказательство. Для получения распределения случайной величины τ рассмотрим следующую последовательность случайных величин

$$T_1 = \tau, \quad T_{k+1} = \min\{t > T_k : \xi_k \le t-k+1, k = 1, \ldots, t\}.$$

Очевидно, что событие $T_{k+1} = n+t$ при условии, что $T_k = n$, возможно тогда и только тогда, когда $\xi_k \le n+2+t-k, k = n+1, \ldots, n+1+t$. Таким образом, для любого k величина $\tau_{k+1} = T_{k+1} - T_k$ не зависит от $T_1, \ldots, T_k$ и имеет то же распределение, что и τ. Другими словами, $\{T_k\}$ - процесс восстановления, а распределение T_k совпадает с распределением суммы k независимых случайных

величин $\tau_1, \ldots, \tau_k$, каждая из которых имеет распределение, совпадающее с распределением τ.

Пусть $f(s) = \sum_{k=1}^{\infty} \mathbf{P}\{\tau = k\} s^k$ - производящая функция распределения τ, и $h_t = \mathbf{P}\{\exists k \geq 1 : t = T_k\}$. Следуя формуле полной вероятности, получим

$$h_t = \sum_{k=1}^{t} \mathbf{P}\{T_k = t\}, t \in \mathbb{N}.$$

Таким образом, если обозначить $h_0 = 1$, получим:

$$H(s) = \sum_{t=0}^{\infty} h_t s^t = 1 + \sum_{t=1}^{\infty} \sum_{k=1}^{t} \mathbf{P}\{T_k = t\} s^t =$$

$$= 1 + \sum_{k=1}^{\infty} \sum_{t=k}^{\infty} \mathbf{P}\{T_k = t\} s^t = 1 + \sum_{k=1}^{\infty} \sum_{t=k}^{\infty} \mathbf{P}\{\tau_1 + \cdots + \tau_k = t\} s^t =$$

$$1 + \sum_{k=1}^{\infty} f^k(s) = \frac{1}{1 - f(s)}.$$

Поскольку t стремится к бесконечности

$$h_t = \mathbf{P}\{\xi_k \leq t - k + 1, k = 1, \ldots, t\} = \prod_{k=1}^{t} \mathbf{P}\{\xi_k \leq k\} \to \prod_{k=1}^{\infty} \mathbf{P}\{\xi_k \leq k\} = q,$$

имеем при $s \uparrow 1$

$$H(s) = \frac{q}{1 - s} (1 + o(1)).$$

С другой стороны, согласно формуле Тейлора, при $s \uparrow 1$, имеем $f(s) = 1 - \mathbf{M}\tau(1 - s) + o(1 - s)$ и при $s \uparrow 1$

$$\frac{1}{1 - f(s)} = \frac{1}{\mathbf{M}\tau(1 - s)} (1 + o(1)).$$

Поэтому $\mathbf{M}\tau = \frac{1}{q}$.

Согласно (6.34) имеем

$$ln\frac{1}{q} + ln(P(\zeta_s \leq 1)) \leq -\sum_{k=2}^{\infty} ln(1 - \frac{1,2}{2^k}) =$$

$$\sum_{k=2}^{\infty}\sum_{j=1}^{\infty}\frac{(\frac{1,2}{2^k})^j}{j} \le 1,2\sum_{k=2}^{\infty}\frac{1}{2^k}+\frac{1}{2}\sum_{k=2}^{\infty}\sum_{j=2}^{\infty}(\frac{1,2}{2^k})^j =$$

$$0,6+\frac{1}{2}\sum_{k=2}^{\infty}\frac{1,44}{2^{2k}}\frac{1}{1-\frac{1,44}{2^k}} \le 0,6+\frac{0,72}{2^4}\frac{1}{1-0,36}\frac{1}{1-\frac{1}{4}} = 0,694.$$

По определению q, применяя (6.35), получим $ln\frac{1}{q} \le 0,694+0,654 = 1.348$, откуда $M(\tau) \le 3,85$.

Если $\theta = k$, то $\tau \le k$. Поскольку $P(\xi_1 = l) = \sigma^l\delta \le \frac{0,72^l}{2}$, из (6.34), получим

$$M(\theta) = \sum_{k=1}^{\infty} kP(\theta = k) =$$

$$\sum_{k=1}^{\infty} k[P(\tau = k)\frac{P(\xi_1 \le k-1)}{P(\xi_1 \le k)} + P(\tau = k-1)\frac{P(\xi_1 = k-1)}{P(\xi_1 \le k-1)}P(\tau = 1)+$$

$$+P(\tau = k-2)\frac{P(\xi_1 = k-2)}{P(\xi_1 \le k-2)}P(\tau = 2)+\cdots+P(\tau = 1)\frac{P(\xi_1 = 1)}{P(\xi_1 \le 1)}P(\tau = k-1)].$$

Замечая, что $\frac{P(\xi_1=l)}{P(\xi_1\le l)} \le 0,72^l$, продолжаем последнее неравенство

$$\le f'(1) - P(\tau = 1)\frac{P(\xi_1 = 1)}{P(\xi_1 \le 1)} + F'(1),$$

где $F(s) = f(0,72s)f(s)$. Поскольку

$$P(\tau = 1) = P(\xi_1 \le 1) \in (0,288+0,288\cdot 0,71; 0,29+\frac{0,72}{2}) = (0,49; 0,65)$$

$$P(\tau = 2) = P(\xi_1 = 2)P(\xi_1 \le 1) \le 0,65\frac{0,72^2}{2} \le 0,1685,$$

то

$$f(0,72) \le 0,72^3 + P(\tau = 1)(0,72-0,72^3) + P(\tau = 2)(0,72^2-0,72^3) \le 0,627$$

$$f'(0,72) \leq 0,72(0,72^2 f'(1) + (1 - 0,72^2)P(\tau = 1) + (0,72 - 0,72^2)P(\tau = 2)) \leq$$

$$0,374 f'(1) + 0,383.$$

Поэтому

$$M(\theta) \leq f'(1) - 0,49 \cdot 0,71 + f'(0,72)0,72 + f'(1)f(0,72) \leq$$

$$\leq f'(1) - 0,347 + 0,27 f'(1) + 0,276 + f'(1)0,627 \leq 1,897 M(\tau) - 0,071 \leq 7,233.$$

Теорема доказана.

6.2.2 Распараллеленные версии блочных алгоритмов с распределёнными операциями

В этом подразделе рассматривается вопрос о том, как оценить время работы программ, реализующих наш алгоритм и алгоритм Видемана-Копперсмита при условии одновременной загрузки нескольких вычислительных узлов [78].

В принципе, каждую операцию любого алгоритма можно рассматривать на предмет возможности использования нескольких однотипных узлов для уменьшения времени на её реализацию. При этом если считать число арифметических операций в рассматриваемом алгоритме фиксированным, то самое значительное снижение времени - это когда время падает обратнопропорционально количеству вычислительных узлов. Так будет, например, при наборе соотношений в методах дискретного логарифмирования и факторизации на основе факторных баз. Однако для большинства относительно более сложных алгоритмов верхняя оценка времени их работы $T(N, d, n, c \ldots; s)$ в зависимости от параметров задачи и используемого оборудования: $N, d, n, c \ldots$, и числа используемых вычислительных узлов s может вести себя по-разному на разных областях изменения аргумента. Зачастую при увеличении s выше

определённого порогового значения $s_0(N, d, n, c \dots)$, зависящего от параметров задачи, время не только не падает обратнопропорционально s, но снова начинает увеличиваться из-за того, что время на обмен между s вычислительными узлами растёт быстрее, чем уменьшается время работы каждого вычислительного узла в отдельности.

По мнению автора, правомерно поставить вопрос о вычислении величин $s_0(N, d, n, c \dots)$ и $T(N, d, n, c \dots) = \min_s T(N, d, n, c \dots; s) = T(N, d, n, c \dots;, s_0(N, d, n, c \dots))$ для некоторого эталонного кластера. В качестве эталона на сегодняшний момент логично взять кластер с неограниченным числом вычислительных узлов, для которых время выполнения одной арифметической операции с машинными словами, умноженное на некоторую константу c, равно времени передачи одного машинного слова между вычислительными узлами. В современной компьютерной технике обмен между оперативной памятью и процессором осуществляется с $c \approx 5$, а между отдельными частями оперативной памяти — $c \approx 20$

Насколько известно автору, параметр c определяется соотношением между тактовыми частотами процессора и «шины», которая связывает вычислительные узлы, процента информационных битов в передаваемых по внутренней сети сообщениях и некоторыми другими характеристиками кластера. На величину c может оказывать влияние место размещения результатов вычислений (ОЗУ, многоуровневый кэш). Будем также предполагать, что на кластере возможна как адресная рассылка между выполняющими конкретные задания вычислительными узлами, так и рассылка с фиксированного узла на узлы некоторой группы по бинарному дереву (бродкаст), то есть за логарифм от числа элементов этой группы. Пересылки внутри непересекающихся групп вычислительных узлов будем считать независимыми. В этом подразделе все оценки времени произведены в единицах, равных времени на производство одной арифметической операции с машинными словами в таком эталонном кластере.

Будем считать также незначительным время на преобразование форматов хранения матриц. Все дальнейшие результаты получены при указанных условиях.

Рассмотрим блочные алгоритмы Видемана-Копперсмита и новый алгоритм. Задачу будем характеризовать четырьмя параметрами: $N \geq 2^{26}$ — размер исходной матрицы (максимум из числа строк и столбцов), d — оценка на число ненулевых элементов (единиц) в каждой строке этой матрицы, n — длина машинного слова, c — параметр, о котором было сказано выше. Таким образом, пересылка V бит будет осуществлена за $\frac{c}{n}V$ единиц времени. Одна единица времени - это время, требуемое для одной арифметической операции с машинными словами.

Распараллеливание первого этапа

Прежде чем перейти к формулировкам теорем, отметим следующее: в последовательных реализациях указанных алгоритмов максимальное время забирает многократно повторяющаяся операция умножения матрицы на блок векторов. Поэтому распараллеливание должно быть прежде всего применено к этой операции.

Можно рассматривать два подхода к распараллеливанию умножения матрицы на блок векторов, связанных с распределённым хранением самой матрицы и (или) распределённым хранением блока.

Необходимость использования первого из этих подходов продиктована также невозможностью хранения в оперативной памяти одного вычислительного узла всей исследуемой матрицы.

Для использования второго подхода рассмотрим блок из ns векторов, а именно будем считать, что во всех рассматриваемых алгоритмах решается система линейных уравнений вида $DX = 0$, где $D \in \mathbb{F}^{N\times N}, X \in \mathbb{F}^{N\times ns}$. Величину s принято называть блочным фактором.

Рассмотрим задачу параллельного распределённого умножения разреженной матрицы на блочный вектор.

Для обеспечения баланса нагрузки необходимо делить разреженную матрицу на части, содержащие приблизительно одинаковое количество единиц. Таким образом, без дополнительных преобразований матрицу можно разрезать только в одном направлении, по соотношениям. Разделим матрицу $D \in \mathbb{F}^{N\times N}$ нашей системы линейных однородных уравнений на горизонтальные полоски по числу l, используемых вычислительных узлов. Таким образом, i-му процессору достанется матрица D_i, в которой ненулевыми оставлены лишь соответствующие N/l строк матрицы D.

На первом этапе нового алгоритма требуется вычислить $(D^TD)^iB$, $B^T(D^TD)^iB, B \in \mathbb{F}^{N\times ns}, i = 1, 2, \ldots$. Тогда

$$D^TD = \sum_{i=1}^{l} D_i^TD_i$$

$$D^TDB_j = \sum_{i=1}^{l} D_i^TD_iB_j, j = 1, \ldots, s, \tag{6.37}$$

где $B_j \in \mathbb{F}^{N\times n}$ - это j-ая вертикальная полоска блока $B \in \mathbb{F}^{N\times ns}$. Для хранения этой полоски и матрицы D_i на соответствующем вычислительном узле потребуется память

$$Nn + \frac{nNd}{l}$$

бит (считаем, что для каждого ненулевого элемента матрицы D_i хранится его номер строки и номер столбца в одном машинном слове).

Таким образом, распределенное вычисление блочного вектора D^TDB на ls вычислительных узлах потребует $\frac{N}{l}d$ операций для вычисления элементов D_iB_j и ещё $\frac{Nd}{l}$, по числу ненулевых элементов в матрице ${D_i}^T$, для завершения вычисления $D_i^TD_iB_j \in \mathbb{F}^{N\times n}$. Отметим здесь, что для второго умножения (на D^T) при такой схеме расчётов не нужны пересылки, поэтому оно занимает существенно меньше времени, чем первое, требующее получения блока B_j.

Далее в алгоритме последовательно вычисляются $(D^TD)^iB$. Для этого требуется пересылка со сложением по формуле (6.37) и обратная рассылка полученной левой части этого равенства на все l задействованых в её вычислении вычислительных узла. Используя алгоритм циклической пересылки [155] для вычисления сложения в формуле (6.37), получим общий объём последовательных пересылок машинных слов со сложением не более $\frac{c}{n}2Nn = 2Nc$. Действительно, в соответствии с алгоритмом циклической пересылки, для получения итоговой суммы на каждом вычислительном узле группы требуется, чтобы этот узел дважды получил и передал блок с размерами не более чем $N \times n$ по двустороннему симметричному каналу связи.

Суммируя, получим итоговую оценку времени на вычисление $(D^TD)B$ на ls вычислительных узлах в виде

$$\frac{2Nd}{l} + 2Nc. \tag{6.38}$$

Вычисление всего пространства Крылова необходимо в новом алгоритме и алгоритме Видемана-Копперсмита для построения решения. Кроме того, на первом этапе рассматриваемых алгоритмов из векторов, составляющих пространство Крылова, строятся коэфициенты рядов в виде скалярных произведений X^TD^iY для алгоритма Видемана-Копперсмита и $B^T(D^TD)^iB$ для рассматриваемого нового алгоритма.

В нашей версии распределённого вычисления векторов $(D^TD)^iB$ на каждом из l вычислительных узлов j-й группы, $1 \le j \le s$, в некоторый момент содержится блочный вектор $(D^TD)^iB_j, i \in \{1, \ldots, k\}$. При вычислении куска пространства Крылова, в случае нового алгоритма, все их можно хранить распределённо на вычислительных узлах группы по $\frac{k}{l}$ на каждом.

Для распределённого по строкам (полоскам) хранения на l вычислительных узлах всего блока B^T потребуется ещё $\frac{Nns}{l}$ бит. После вычисления $(D^TD)^iB_j$ на каждом из l вычислительных узлов j-й группы вычисляется своя часть

скалярного произведения $B^T(D^TD)^iB_j$ не более, чем за

$$3\frac{Nns}{l(log_2\frac{N}{2}-log_2log_2\frac{N}{2})}$$

арифметических операций. При $N \geq 2^{26}$, откуда $\frac{N}{2} \geq N^{\frac{10}{13}}log_2\frac{N}{2}$, эту дробь можно оценить сверху величиной

$$\frac{4Nns}{l \cdot log_2N}. \tag{6.39}$$

Все части после конкатенации внутри одной группы дают $B^T(D^TD)^iB_j$, а по всем группам - $B^T(D^TD)^iB$. Пересылки при этом незначительные ввиду маленьких размеров пересылаемых матриц. Добавляя к оценке (6.38), получим оценку времени на вычисление очередного блока векторов из пространства Крылова и коэффициента ряда на кластере из sl вычислительных узлов:

$$\frac{2dN}{l}+\frac{4Nns}{l \cdot log_2N}+2Nc. \tag{6.40}$$

Более детальные вычисления показывают, что при актуальных на сегодняшний день значениях параметров, например таких, что $2ns < dlog_2N$, второе слагаемое оценки (6.40) меньше третьего. Для больших значений s скалярные произведения лучше вычислять на дополнительных вычислительных узлах таким образом, чтобы время, соответствующее второму слагаемому в равенстве (6.40), было меньше первого и третьего.

Дадим несколько комментариев о дополнительных возможностях распараллеливания вычисления скалярных произведений.

Будем вычислять скалярные произведения на p дополнительных к уже рассмотренным l вычислительных узлах в группе. То есть каждая из s групп расширяется на p вычислительных узлов, на которых хранится $\frac{1}{p}$-ая часть строк блока B^T и в буфере r последовательных результатов (достаточно будет один

или два) вычислений по формуле (6.37). Для этого потребуется память порядка

$$\frac{Nns}{p} + rNn, \text{или}, \frac{Nn}{8 \cdot 10^9}\left(r + \frac{s}{p}\right) \text{ГВ}.$$

На эти узлы по бинарному дереву пересылаются результаты вычислений по формуле (6.37) за время

$$\frac{c}{n} Nnlog_2 p = Nclog_2 p.$$

Время на пересылку с последующей конкатенацией скалярных произведений мало ввиду малого их размера, и мы его учитывать не будем.

Загрузка в новом алгоритме каждого из p вычислительных узлов своей частью левого множителя B^T производится адресной рассылкой один раз за весь алгоритм. Оценку времени на вычисление скалярных произведений получаем аналогично оценке (6.39) в виде

$$\frac{4Nns}{plog_2 N},$$

и далее, аналогично оценке (6.40) получаем оценку времени на вычисление очередного блока векторов из пространства Крылова и очередного коэффициента ряда на кластере из

$$s(l+p) \tag{6.41}$$

вычислительных узлов

$$max\left(\frac{2dN}{l} + 2Nc; Nclog_2 p + \frac{4Ns}{plog_2 N}\right)$$

Для упрощения дальнейших расчётов уравняем два слагаемых во втором элементе, положив

$$p \approx \lceil \frac{4s}{clog_2 N} \rceil. \tag{6.42}$$

При таком выборе и актуальных на сегодняшний день значениях параметров время на скалярные произведения будет меньше, чем время на умножения матрицы на блок, что позволит иметь минимальный буфер в узлах, вычисляющих скалярные произведения (то есть $r = 1$ или 2).

При достаточно больших l получится оценка

$$Nc \cdot max\left(2; log_2\lceil\frac{4s}{clog_2N}\rceil\right) \tag{6.43}$$

с общей оценкой по всему первому этапу из $2\frac{N}{ns}$ шагов

$$\frac{2N^2c}{ns}max\left(2; log_2\lceil\frac{4s}{clog_2N}\rceil\right) \tag{6.44}$$

с условием $s(l+p) \leq S$, или

$$s\left(l + \frac{4s}{clog_2N}\right) \leq S,$$

где S - общее число вычислительных узлов кластера.

Отметим, что в данном случае мы предполагаем, что s рассматриваемых групп не связаны между собой сетью. В противном случае выгоднее использовать формулу $\alpha_{i+j} = B^T A^{i+j} B = (A^i B)^T A^j B$, что уменьшает число умножений на A до $\frac{N}{ns}$, требуя при этом обмена частями блоков $A^i B$. При этом, чем больше можно запасти блоков $A^i B$, тем реже нужно осуществлять обмен.

При этом, оперативная память одного узла, занимающегося умножением матрицы на блок, должна позволять размещать данные порядка

$$\underset{\text{Матрица } D_i}{\frac{nNd}{l}} + \underset{\text{Текущее } B_j}{Nn} + \underset{\text{Часть } B^T}{\frac{Nns}{l}} = \frac{Nn}{8\cdot 10^9}(1 + \frac{d+s}{l})\text{ГВ}, \tag{6.45}$$

В случае $N \approx 2^{26}$ — это составляет примерно

$$\frac{1}{2}(1 + \frac{d+s}{l})\text{ГВ}. \tag{6.46}$$

К этому ещё надо добавить объём, необходимый на каждом узле для хранения отрезка длины k (об этом параметре речь пойдёт ниже) базиса пространства Крылова, то есть $\frac{Nn}{l}k$.

Отметим, что если скалярные произведения вычисляются на дополнительных узлах, то требования к памяти одного узла, занимающегося умножением матрицы на вектор, уменьшаются до

$$\frac{1+c}{2}\text{ГБ}.$$

Узлы, занимающиеся вычислением скалярных произведений, должны иметь следующий объём оперативной памяти

$$\underset{\text{Часть } B^T}{\frac{Nns}{p}} + \underset{\text{Текущее по } i \text{ значение } (D^TD)^iB_j}{Nn} = \frac{Nn}{8\cdot 10^9}\left(\frac{s}{p}+1\right)\text{ГВ}, \tag{6.47}$$

При подсчёте скалярных произведений на тех же узлах, на которых вычисляется умножение матрицы на вектор для оценки памяти надо сложить оценки (6.45) и (6.47).

В случае алгоритма Видемана-Копперсмита k растёт до $2\frac{N}{ns}$, и мы не имеем возможности хранить все вектора пространства Крылова. Оценка на память в этом случае задаётся выражением (6.46). Оценку времени получим аналогично (6.40), где отсутствует коэффициент 2 при d, так как нет D^T. Значение

$$l = \frac{d}{2c}.$$

для времени всего первого этапа, содержащего вычисление $\frac{2N}{ns}$ коэффициентов ряда, с учётом необходимости двух проходов, (для построения коэффициентов ряда и для построения решения), даёт близкие к предыдущим оценки:

$$\frac{16N^2c}{sn}. \tag{6.48}$$

Вычисление скалярных произведений можно построить иначе. А именно, после

того, как на каждом из l вычислительных узлов j-й группы, $1 \leq j \leq s$ был получен блочный вектор A^iB_j, в обозначениях только что доказанной теоремы, сделаем следующее. Пусть $(A^iB_j)_r \in \mathbb{F}^{\frac{N}{l} \times n}, r \in \{1, \ldots, l\}$ обозначает $\frac{N}{l}$ строк блока A^iB_j, хранящиеся на r-м узле j-й группы. Проведём обмен с целью вычисления на этом же узле элементов

$$(A^iB_{j_1})_r^T(A^iB_j)_r, \quad (A^{i-1}B_{j_1})_r^T(A^iB_j)_r$$

для $j_1 \in \{1, \ldots, s\}$ таких, что $(-1)^{j-j_1} sign(j - j_1) = 1$. Тем самым после относительно простого суммирования все элементы $B_{j_1}^T A^{2i} B_{j_2}, B_{j_1}^T A^{2i-1} B_{j_2}, j_1, j_2 \in \{1, \ldots, s\}$ будут вычислены. Обмен потребует

$$\frac{s}{2} \cdot \frac{c}{n} \cdot \frac{Nn}{l}$$

единиц времени, а счёт

$$2\frac{4Nn}{l \cdot log_2 2\frac{N}{l}} \tag{6.49}$$

соответственно. Суммируя с (6.38), получим

$$\frac{2Nd}{l} + 2Nc + \frac{scN}{2l} + \frac{8Nn}{l \cdot log_2 2\frac{N}{l}}$$

В этой оценке по сранению с (6.40) есть дополнительное слагаемое, которое отвечает за пересылки на кластере, но выбором $l \geq max\left(s, \frac{2d}{c}, \frac{16n}{c}\right)$ можно добиться, чтобы все слагаемые кроме второго были достаточно малы. При этом число шагов всего алгоритма будет $\frac{N}{ns}$, то есть в 2 раза меньше и общую оценку времени работы первого этапа можно взять

$$\frac{2N^2c}{ns}. \tag{6.50}$$

При такой системе расчёта хранить части B^T не нужно и общая необходимая память одного узла аналогично (6.45) (6.46) оценивается величиной

$$\frac{Nn}{8 \cdot 10^9}(1 + \frac{d+k}{l})\text{ГВ},$$

что при $N \approx 2^{26}$ составляет

$$\frac{1}{2}(1 + \frac{d+k}{l})\text{ГВ}.$$

Распараллеливание второго этапа

В алгоритме Видемана в версии Копперсмита вектор $\bar{g}$ строится, последовательно увеличивая его размер. В работе [148] предложен другой вариант, имеющий практически линейную оценку сложности, правда, с несколькими логарифмическими сомножителями и достаточно большими мультипликативными константами. Оптимизированный таким способом алгоритм будем называть алгоритмом Видемана-Копперсмита с матричным умножением. Остановимся на этом подробнее. Пусть

$$h = h(\lambda) = \sum_{i=0}^{\infty} H_i \lambda^i, H_i \in \mathbb{F}^{ns \times ns}. \tag{6.51}$$

Здесь размер ns выбран так, чтобы соответствующий алгоритм можно было применить к построению $\bar{g}$ на втором этапе алгоритма Видемана-Копперсмита с использованием блочного фактора s.

Степенью векторного многочлена $m(\lambda) \in (\mathbb{F}[\lambda])^{2ns \times 1}$ называется его степень как многочлена с векторными коэффициентами из $\mathbb{F}^{2ns \times 1}$.

Порядком векторного многочлена $m(\lambda) \in (F[\lambda])^{2ns \times 1}$ называется целое неотрицательное j, такое, что $h(\lambda)m(\lambda) = \sum_{i \geq j+1} \mu_i \lambda^i, \mu_{j+1} \neq \overline{0}$.

σ-базисом ряда h назовём матричный многочлен $M(x) \in \mathbb{F}^{2ns \times 2ns}[\lambda]$, удовлетворяющий следующим условиям:

1. Столбцы $M^{(i)}(\lambda)$ матрицы $M(\lambda)$ имеют порядок не меньше σ.

2. Для любого $v \in F^{2ns \times 1}[\lambda]$, имеющего порядок не меньше σ, существует единственное представление в виде

$$v = \sum_{i}^{2ns} M^{(i)} C^{(i)}, C^{(i)} \in \mathbb{F}[\lambda], degC^{(i)} + degM^{(i)} \leq degv.$$

Для реализации алгоритма Видемана-Копперсмита необходимо построить приближения Паде к ряду вида (6.51), где $H_i = X^T D^i Y; X, Y \in \mathbb{F}^{N \times ns}$, а именно такие $P(\lambda), Q(\lambda) \in \mathbb{F}^{ns \times 2ns}[\lambda]$, что

$$(h \| I_{ns}) \begin{pmatrix} Q \\ P \end{pmatrix} = O(\lambda^{2d+1}), degQ, degP \leq d, \tag{6.52}$$

где I_{ns} - единичная матрица из $\mathbb{F}^{ns \times ns}, d = \frac{N}{ns}$.

Положим в дальнейшем, что числа N, n, s являются степенями двойки.

Теорема 6.7. *[78] Оценка времени работы параллельной реализации с использованием блочного фактора второго этапа алгоритма Видемана-Копперсмита с матричным умножением при* $N \geq 2^{26}$

1) При

$$s \geq \frac{120 log_2(log_2 32N)(log_2 \frac{ns}{2} - log_2 log_2 \frac{ns}{2})}{17n} \tag{6.53}$$

имеет вид:

$$1122 N n^2 s^3 log_2 32N log_2 4N \tag{6.54}$$

2) При

$$s \leq \frac{120 log_2(log_2 32N)(log_2 \frac{ns}{2} - log_2 log_2 \frac{ns}{2})}{17n} \tag{6.55}$$

имеет вид:

$$15840 N n s^2 log_2(32N) log_2(4N) log_2(log_2(32N)) \tag{6.56}$$

Замечание. *Так как из неравенства* $z \geq 2Alog_2 A$ *следует неравенство* $z \geq Alog_2 z$, *то при* $s \geq \frac{120 log_2(log_2 32N)}{17n} log_2 \left(\frac{120 log_2(log_2 32N)}{34n} \right)$ *неравенство (6.53) выполнено.*

Доказательство теоремы 6.7.

Для построения необходимого приближения будем использовать алгоритм из статьи [148], который строит σ-базисы, последовательно удваивая σ. Для того, чтобы каждый раз был построен весь базис, в этой статье осуществляется переход от ряда $h(\lambda)$ к ряду $h(\lambda^{2ns}) \times (1, \lambda, \lambda^2, \ldots, \lambda^{2ns-1})^T$, к которому строятся приближения с порядком уже $2dns$, используя $2d$ коэффициентов ряда $h(\lambda)$. Оценку сложности работы $C(ns, ns, 2dns)$ соответствующего алгоритма можно получить из доказательства теоремы 2.4 [148] с заменой d на $2dns$, а именно

$$C(ns, ns, 2dns) \leq 2C(ns, ns, dns) + MM(ns, dns) + MM(ns, 2dns),$$

где $MM(a, b)$ - сложность умножения двух матричных многочленов степени b из $\mathbb{F}^{a \times a}[\lambda]$. Продолжая аналогично, получим

$$\begin{gathered} C(ns, ns, 2dns) \leq \\ MM(ns, 2dns) + \sum_{i=1}^{log_2 2dns} MM(ns, 2^{-i}2dns)(2^i + 2^{i-1}) \leq \\ \tfrac{3}{2} \sum_{i=0}^{log_2 2dns} 2^i MM(ns, 2^{-i}2dns) \end{gathered} \tag{6.57}$$

Напомним, что мы предполагаем параметры N, n, s, равными степеням двойки.

Оптимизированный алгоритм умножения матричных полиномов предложен в статье [160]. Согласно лемме 3.2 из этой работы, верна следующая оценка

$$MM(ns, 2^i) \leq \alpha_Q \frac{(ns)^2}{n} + \beta_Q \frac{(ns)^2 \frac{ns}{n}}{log_2 \frac{ns}{2} - log_2(log_2(\frac{ns}{2}))}.$$

где $Q : \varphi(r^Q) \geq 2(2^i + 1), H : 2^H + 2 \leq Q \leq 2^{H+1} + 1$, $\alpha_Q \leq r^Q 2^H((6r + 2)\mu_r(H + 1) + \frac{2\alpha_2}{r^2})$, $\beta_Q \leq r^Q 2^H \beta_2$, где в качестве r может быть выбрано любое натуральное число, отличное от 2, μ_r - сумма модулей коэффициентов в круговом многочлене с номером r, а α_2 и β_2 соответственно количества сложений и умножений для умножения многочленов степени $\varphi(r^2) - 1$. Известно, что при $r = 3$ можно выбрать β_2, равным 17. Здесь применён обычный алгоритм сложения матриц,

разбитых по строкам на машинные слова длины n, а также оптимизированный алгоритм умножения таких матриц, изложенный выше. При $r = 3$ получим $\mu_r = 3$. Q может быть выбрано наименьшим таким, чтобы $3^{Q-1} \geq 2^i + 1$ (см. стр. 8 [148]), поэтому $3^Q \leq 9(2^i + 1)$, откуда при $i \geq 1$ выполнено $3^Q \leq 2^{i+4}$, и, поскольку $3log_3 2 < 2$, имеем:

$$M(ns, 2^i) \leq$$
$$\frac{(ns)^2}{n} 2^{i+4} log_3 2^{i+4} \left(20 \cdot 3(log_2 log_3 2^{i+4} + 1) + \frac{2}{9}\alpha_2 + \frac{17ns}{nlog_2 \frac{ns}{2} - log_2 log_2 \frac{ns}{2}} \right) \leq$$
$$\frac{(ns)^2}{n} \frac{2^{i+5}}{3} (i+4) \left(60(log_2 \frac{2(i+4)}{3} + 1) + \frac{2}{9}\alpha_2 + \frac{17ns}{log_2 \frac{ns}{2} - log_2 log_2 \frac{ns}{2}} \right).$$

Поскольку неравенство

$$\frac{2}{9}\alpha_2 + 60 log_2 \frac{4(i+1)}{3} \leq 120 log_2 log_2 32dns,$$

очевидно, выполнено и $i \leq log_2 2dns$, рассматриваемую оценку можно продолжить величиной

$$ns^2 2dns 11(log_2 2dns + 4) \left(120 log_2 log_2 32dns + \frac{17ns}{log_2 \frac{ns}{2} - log_2 log_2 \frac{ns}{2}} \right). \qquad (6.58)$$

Таким образом,

$$C(ns, ns, 2dns) \leq$$
$$\frac{3}{2} ns^2 22 \cdot 2dns(log_2 2dns + 4) log_2 4dns \left(120 log_2 log_2 32dns + \frac{17ns}{log_2 \frac{ns}{2} - log_2 log_2 \frac{ns}{2}} \right) \leq$$

$$66Nns^2 log_2 32N log_2 4N \left(120 log_2 log_2 32N + \frac{17ns}{log_2 \frac{ns}{2} - log_2 log_2 \frac{ns}{2}} \right). \qquad (6.59)$$

В случае 1) это неравенство можно продолжить величиной

$$\frac{132Nns^2log_232Nlog_24N \cdot 17ns}{log_2\frac{ns}{2} - log_2log_2\frac{ns}{2}}.$$

Поскольку в этом случае $\frac{ns}{2} \geq 16$, то эту оценку можно продолжить величиной

$$1122Nns^2log_2(32N)log_2(4N).$$

Пункт второй очевидно следует из оценки (6.59). Теорема доказана.

Теорема 6.8. *[78] Оценка времени работы параллельной реализации с использованием блочного фактора алгоритма Видемана-Копперсмита с матричным умножением при* $N \geq 2^{26}$ *и*

$$N > \frac{71log_2(32N)log_2(4N)}{cn}\left(\frac{240log_2(log_2(32N))log_2(\frac{60}{17n}log_2(log_2(32N)))}{17}\right)^4$$

имеет вид

$$94N^{1+\frac{3}{4}}n^{-\frac{1}{4}}c^{\frac{3}{4}}(log_2(32N)log_2(4N))^{\frac{1}{4}}.$$

Доказательство. На третьем этапе алгоритма строится решение как линейная комбинация базисных векторов пространства Крылова с известными коэффициентами. Поскольку эта часть легко параллелизуется (по строкам векторов базиса в соответствии с их хранением, см. распараллеливание первого этапа) и время на эту операцию может быть сделано незначительным и мы его учитывать не будем. Точная оценка приведена в доказательстве следующей теоремы.

Доказательство теоремы следует из теоремы 6.7 пункт 1), замечания к ней и оценки (6.48) при

$$s = \frac{1}{4}\left(\frac{8Nc}{561n^3log_2(32N)log_2(4N)}\right)^{\frac{1}{4}}.$$

Заметим, что при $N \geq 2^{26}, c \geq 1$, оценка из этой теоремы выполнена.

6.2.3 Итоговые оценки и тесты

Для удобства читателя сгруппируем лучшие верхние оценки для разных алгоритмов в таблицу

	Время
Wiedemann-Coppersmith with matrix polynomial multiplication	$94N^{1+\frac{3}{4}}n^{-\frac{1}{4}}c^{\frac{3}{4}}((log_2 32N)(log_2 4N))^{\frac{1}{4}}$
Wiedemann-Coppersmith-Thom*é*([161], (6.48))	$O(N^{1+\frac{2}{3}}(log_2 N log_2 log_2 N)^{\frac{1}{3}})$
Montgomery (1995) Алгоритм-K	$\frac{2c}{n}N^2$

Отметим, что предложенный подход позволил сконструировать целую серию алгоритмов, которые могут иметь преимущества перед известными алгоритмами в случае использования вычислителя специфической структуры и мощности. Возможны и дальнейшие упрощения предложенной процедуры, что повлияет на сокращение времени работы всего алгоритма.

Кроме того, изложенный подход фактически связывает между собой алгоритмы, основанные на ортогонализации пространства Крылова и алгоритмы, основанные на поиске соотношения для рекуррентной последовательности. Давно замечено, что реализации этих подходов имеют практически одинаковую

эффективность. Изложенный подход позволяет понять теоретические причины этого явления и использовать преимущества обоих методов.

В открытой части отчёта по НИР Академии криптографии за 2010 год [76] с.н.с. Института вычислительной математики РАН к.ф.-м.н. Н.Л.Замарашкин провёл анализ написанных им версий алгоритмов Монтгомери и автора настоящей диссертации. Он показал, что для тестируемых матриц текущая упрощённая реализация метода автора настоящей работы превосходит по эффективности существующую реализацию метода Монтгомери (2008) в три раза. Затем он описал модификацию текущей реализации метода Монтгомери, которая, на данный момент, по эффективности сопоставима с реализацией моего метода; показывается, однако, что параллельный ресурс (возможность время за счёт добавления вычислительных узлов) метода Монтгомери в значительной степени исчерпан, в то время как для моего метода это совсем не так.

Программный комплекс реализован в виде набора модулей, написанных в стандарте ANSI C и с использованием библиотеки межпроцессорного обмена сообщениями MPI. В качестве тестовых использовались три матрицы. В частности, использовалась матрица «m512t» размером $5\ 452\ 543 \times 5\ 768\ 669$ (число строк и столбцов) и средним числом ненулевых элементов в строках — 51.

При работе над данным проектом эксперименты проводились на следующих вычислительных системах: суперкомпьютере СКИФ МГУ «Чебышев», кластере МВС-1000.

Время работы алгоритма Монтгомери для «легкой» матрицы «m512t» около 50 часов.

Хотя для упрощения программирования алгоритм, предложенный автором, был изменён, что привело по мнению Н.Л.Замарашкина, как минимум, к удвоению времени, общее время его работы составило 17 часов («m512t», МВС-1000).

6.2.4 Дальнейшие исследования

Новый алгоритм типа Видемана-Копперсмита

В этом подразделе рассматривается версия нового алгоритма, построенная по принципу алгоритма Видемана-Копперсмита [77]. Он использует меньшую оперативную память, чем алгоритм Видемана-Копперсмита. При этом, выбором большего значения блочного фактора (коэффициент пропорциональности между шириной рассматриваемых блоков и длиной машинного слова) при том же объёме используемой оперативной памяти может быть выигрыш по времени в сравнении с алгоритмом Видемана-Копперсмита.

Предпочтения алгоритма Видемана-Копперсмита по сравнению с алгоритмом Монтгомери при постановке рекорда факторизации объясняются тем, что первый позволяет проводить большую часть вычислений на совершенно независимых вычислительных системах. Алгоритм разбивается на три этапа:

- Вычисление коэффициентов рядов.
- Вычисление приближений к этим рядам.
- Вычисление решения с помощью коэффициентов этих приближений.

Время, необходимое для первого и третьего этапа обычно выбирают значительно большим, чем для второго. Это связано с тем, что второй этап требует использования вычислителя с большим объёмом оперативной памяти. Ниже предложена версия нового алгоритма, которая требует меньшей оперативной памяти, чем на втором этапе алгоритма Видемана-Копперсмита с учётом предложений Томе [161]. Заменяя в алгоритме параметр n на ns, мы уменьшаем в s раз число шагов, так как пространство Крылова будет состоять уже из меньшего числа блоков большей ширины, но увеличиваем объём используемых многочленов, так как матрицы их коэффициентов растут в s раз

по обоим направлениям. Поэтому, несмотря на то, что их степень уменьшается в s раз, объём памяти для их хранения увеличивается также в s раз.

Обозначим, как и раньше, d верхнюю границу числа ненулевых элементов в каждой строке матрицы линейной системы, ns - число векторов в используемых блоках. В этих обозначениях наша цель — получить нетривиальное решение системы

$$DX = B, D \in \mathbb{F}^{M \times N}, M \leq N; B, X \in \mathbb{F}^{N \times ns}. \tag{6.60}$$

Как в предыдущем разделе, обозначим c отношение между временем пересылки одного машинного слова между узлами и временем, за которое процессор производит одну операцию с машинными словами.

Как было отмечено в предыдущем разделе, время на одно умножение на разреженную матрицу примерно оценивается величиной $4Nc$, а число таких умножений $2\frac{N}{ns}$ на первом этапе, и $\frac{N}{ns}$ на третьем [162] (оригинальный алгоритм Видемана-Копперсмита требует $2\frac{N}{ns}$ матричных умножений на 3 этапе). Эта оценка получена для вычислительных систем из $s\frac{d}{2c}$ вычислительных узлов и для достаточно большого d при рассмотрении алгоритма Видемана-Копперсмита.

Поскольку в алгоритмах [72] и [145] умножение на разреженную матрицу заменено на последовательное умножение на матрицы D и D^T (см. [78]), оптимальное число вычислительных узлов, которое позволяет приравнять время на счёт и время на пересылки, будет равно $s\frac{d}{c}$.

Отметим, что, несмотря на двойное умножение, метод, изложенный в [78], даёт одинаковые оценки на время при одиночном и двойном умножении при использовании во втором случае большего количества вычислительных узлов.

Число арифметических опрераций на 2 этапе алгоритма Видемана-Копперсмита-Томе оценено в [161] величиной

$$O(Nns(ns + log_2 N)log_2 N log_2 log_2 N).$$

Складывая с (6.48) и оптимизируя по s, получаем для всего алгоритма оценку

$$O(N^{1+\frac{2}{3}}(log_2 N log_2 log_2 N)^{\frac{1}{3}}).$$

Соответствующее время работы параллельной программы реализации 2 этапа алгоритма на кластере с объёмом оперативной памяти около 1 ТВ, при $N \approx 1,9 \cdot 10^8, s = 8, n = 64$, оказалось равным 17,3 часов ([162]).

Как и в версии типа Монтгомери, на первом этапе вычислим приблизительно $2\frac{N}{ns}$ коэффициентов ряда:

$$\alpha = \sum_{i=0}^{\infty} \alpha_i \lambda^{-i}, \alpha_i = B^T A^i B, A = D^T D. \tag{6.61}$$

На 2 этапе построим приближения к ряду (6.61). Для этого можно использовать технику, взятую из оптимизированной версии. Для простоты изложения рассмотрим вариант с техникой из первой версии нового алгоритма. Сначала для некоторого t_0, выбор которого разберём позже, для каждого $t \in \{1, \ldots, t_0\}$ вычислим $Q^{(t)}(\lambda)$ такой, что

$$\alpha(\lambda)Q^{(t)}(\lambda) - P^{(t)}(\lambda) = \sum_{i=t+1}^{\infty} \rho_i^{(t)} \lambda^{-i}, degQ^{(t)}(\lambda) \leq t, degP^{(t)}(\lambda) \leq t,$$

при помощи матрицы

$$\begin{pmatrix} \alpha_{t+1} & \ldots & \alpha_2 & \alpha_1 \\ \alpha_{t+2} & \ldots & \alpha_3 & \alpha_2 \\ \ldots & \ldots & \ldots & \ldots \\ \alpha_{2t} & \ldots & \alpha_{t+1} & \alpha_t \\ \lambda^t Q^{(0)} & \ldots & \lambda Q^{(0)} & Q^{(0)} \\ I_n & \ldots & O_n & O_n \\ \ldots & \ldots & \ldots & \ldots \\ O_n & \ldots & I_n & O_n \\ O_n & \ldots & O_n & I_n \end{pmatrix} \begin{pmatrix} \upsilon_t \\ . \\ . \\ . \\ \upsilon_1 \\ \upsilon_0 \end{pmatrix} = \begin{pmatrix} O_n \\ . \\ . \\ O_n \\ Q^{(t)} \\ Q_{t+1}^{(t)} \\ . \\ . \\ Q_0^{(t)} \end{pmatrix}. \tag{6.62}$$

Верхняя часть матрицы в левой части имеет размер t на $t+1$ блоков из $\mathbb{F}^{ns\times ns}$. Таким образом, соответствующая система линейных однородных уравнений имеет решение в виде блока полного ранга: $\bar{v} \in \mathbb{F}^{(t+1)ns\times ns}$, а полученная матричная аппроксимация $Q^{(t)}(\lambda)$ имеет полный ранг над полем $\mathbb{F}$.

Для любого $Q^{(i)}(\lambda)$ получим $\tilde{Q}^{(i)}(\lambda) = \sum_{j=0}^{i} \tilde{Q}_j^{(i)}\lambda^j, \tilde{Q}_j^{(i)} \in \mathbb{F}^{ns\times ns}$, где $\tilde{Q}_i^{(i)}$ - невырождена и

$$\alpha(\lambda)\tilde{Q}^{(t)}(\lambda) - \tilde{P}^{(t)}(\lambda) = \sum_{i=t+1-\delta(t)}^{\infty} \tilde{\rho}_i^{(t)}\lambda^{-i}, deg\tilde{Q}^{(t)}(\lambda) \le t, deg\tilde{P}^{(t)}(\lambda) \le t,$$

для не очень большого натурального $\delta(t)$, как это было описано в параграфе, посвящённом алгоритму типа Монтгомери. Продолжая аналогичные вычисления, получаем $Q^{(t)}(\lambda), \tilde{Q}^{(t)}(\lambda), t > t_0$ при помощи $Q^{(i)}(\lambda), \tilde{Q}^{(i)}(\lambda), i \in \{t-1, \ldots, t-\theta(t)\}$. Исходя из результатов численного эксперимента, верхняя оценка параметра $\theta(t)$, обеспечивающая успешную работу алгоритма с вероятностью не менее $0,99$, растёт пропорционально $log\frac{N}{n}$. Таким образом, t_0 мы можем выбирать вблизи этой границы.

Если обозначить

$$U_i = \tilde{Q}^{(i)}(A, B) = \sum_{j=0}^{i} A^j B\tilde{Q}_j^{(i)} \in \mathbb{F}^{N\times ns}, \tag{6.63}$$

как и в алгоритме типа Монтгомери, получим:

$$U_i^T A U_t = \sum_j \tilde{Q}_j^{(i)T}\tilde{\rho}_{j+1}^{(t)}. \tag{6.64}$$

Поэтому, если $i+1 < t+1-\delta(t)$, это равняется нулю. Согласно численному эксперименту, матожидание параметра $\delta(t)$ не превышает 2. Поэтому мы можем быстро ортогонализовать блоки U_i и сформировать из их столбцов блоки $W_i, i = 1, \ldots, m-1$, перенося некоторые из них в следующий этап. Тогда формула

$$X = \sum_{i=0}^{m-1} W_i (W_i^T A W_i)^{-1} W_i^T B \tag{6.65}$$

позволит построить решение системы также, как и в алгоритме Монтгомери [145]. Скалярные произведения $W_i^T A W_i$ мы получим из (6.64). Как и в алгоритме типа Монтгомери,

$$U_i = A U_{i-1} C_{i-1} + U_{i-1} c_{i-1} + \cdots + U_{i-\theta(i)+1} c_{i-\theta(i)+1} + V_{i-\theta(i)} c_{i-\theta(i)},$$

для некоторых $C_i, c_i \in \mathbb{F}^{ns \times ns}$, где $V_k = Q^{(k)}(A, B)$, и $W_i = \sum_{k=0}^{\Delta(i)} U_{i-k} \delta_k, \delta_k \in \mathbb{F}^{ns \times ns}, \Delta(i) \approx \delta(i), B = U_0$. Таким образом, элементы $W_i^T B$ могут быть получены при помощи рекуррентных формул

$$U_i^T B = C_{i-1}^T U_{i-1}^T A B + \sum_{k=1}^{\theta(i)-1} c_{i-k}^T U_{i-k}^T B + c_{i-\theta(i)}^T V_{i-\theta(i)}^T B.$$

На 2 этапе нашего алгоритма заменим вектора U_i из (6.63) совокупностью их коэффициентов $\tilde{Q}_j^{(i)}$ в базисе $A^j B$, где $j = 0, 1, \ldots, i$. Аналогично мы можем заменить вектора W_i и X. Обозначив соответствующие коэффициенты X как X_i, на 3 этапе нашего алгоритма получим

$$X = \sum_{j=0}^{j_0} A^j B X_j, j_0 \approx \frac{N}{ns},$$

из блочных векторов $A^j B$, которые мы должны вычислить во второй раз. Отметим, что в соответствующем месте алгоритма Видемана-Копперсмита $j_0 \leq 2\frac{N}{ns}$.

Оценка времени для 1 и 3 этапа такого алгоритма примерно такая же, как в алгоритме Видемана-Копперсмита. Эти этапы могут быть вычислены на s несвязанныъх кластерах, имеющих достаточную опративную память для хранения матрицы A.

Время работы 2-го этапа нашего алгоритма может быть приблизительно оценено числом арифметических операций по формуле:

$$O(\sum_{i=1}^{\frac{N}{ns}} i \cdot ns \cdot ns \cdot s) = O(N^2 s). \tag{6.66}$$

Эти вычисления проводятся без рекурсии и умножения матричных многочленов. Это линейные комбинации блоков из $\mathbb{F}^{ins \times ns}$ с коэффициентами из $\mathbb{F}^{ns \times ns}$, умножения блоков из $\mathbb{F}^{ins \times ns}$ друг на друга и некоторые другие сравнительно более быстрые операции, такие как произведения и суммы матриц из $\mathbb{F}^{ns \times ns}$. Исходя из результатов численного эксперимента, константа в формуле (6.66) невелика (примерно 5), и при помощи не очень сложного распараллеливания с небольшой коммутацией мы можем получить время, совпадающее с (6.66), делённым на число вычислительных узлов, которые имеют общую оперативную память объёмом, указанным в оценке, полученной выше. Как было отмечено в предыдущем разделе, оптимальное значение вычислительных узлов для 1 и 3 этапа равняется $O(s^2)$, где $s = O(N^{\frac{1}{3}})$. Использование такого количества вычислительных узлов на 2 этапе делает время его работы равным времени работы 1 и 3 этапов.

Связь между приближениями рядов по положительным и отрицательным степеням формальной переменной

В данном подразделе изучаются свойства алгоритма Видемана-Копперсмита. В частности, для случая симметричной системы линейных уравнений из приближений формального ряда, которые строятся в этом алгоритме в шагах с нечётным номером, построен ортогональный базис пространства Крылова. Предложены модификации алгоритма, использующие эти свойства. Эти результаты опубликованы в [80]

В настоящее время для решения больших разреженных систем линейных

уравнений над полем из двух элементов, чаще всего, используются блочные алгоритмы Монтгомери и Видемана-Копперсмита. Принципиальное отличие между ними в том, что алгоритм Монтгомери последовательно строит и ортогонализует пространство Крылова, одновременно, выписывая в нём координаты вектора, образ которого, при действии матрицы рассматриваемой системы, ортогонален всему пространству Крылова. Алгоритм Видемана-Копперсмита строит фактически такой же вектор в виде координат приближения некоторого ряда по положительным степеням формальной переменной. Первоначальная последовательная версия этого алгоритма была ускорена в работе Томэ [161] ценой увеличения объёма используемой оперативной памяти, что на практике (см. [162]) сковывает возможности выбора параметров, обеспечивающих быстрейшее выполнение всего алгоритма.

В данном разделе выявлены некоторые общие принципы, на которых работают алгоритмы Монтгомери и Видемана-Копперсмита. Представляется, что это поможет воспользоваться преимуществами каждого из них.

Заметим, что в алгоритме Видемана-Копперсмита, также как и в алгоритмах Ланцоша и Монтгомери, решение находится через его координаты в пространстве Крылова. Отметим также, что в общем случае теоретическая оценка снизу [147] на вероятность получения решения этим алгоритмом равна $1-(1-0,03)^{n'}$. Нетрудно видеть, что коэффициенты g_i (см. приложение) решения в пространстве Крылова, ищутся в шаге 3 как коэффициенты некоторого достаточно хорошего приближения ряда по положительным степеням некоторой формальной переменной. Их можно искать, последовательно наращивая порядок приближения, как это сделано Копперсмитом в работе [146], или при помощи сравнительно небольшого количества приближений с порядками, равными степеням двойки, как предлагается в работах [148, 157]. Отметим, что в этих работах строится базис всех приближений при помощи асимптотически более быстрого алгоритма с количеством операций $O(Nlog^2NloglogN)$, против

$O(N^2)$ у Копперсмита. Однако константа в O в первой оценке достаточно большая, кроме того, она плохо зависит от так называемого блочного фактора, ускоряющего весь алгоритм при использовании параллельных вычислений. Для постановки последнего рекорда разложения чисел RSA на множители (12 декабря 2009 года разложено число RSA-768, содержащее 232 десятичные цифры или 768 бит) была взята [161] аналогичная процедура как в исходной версии Копперсмита, но строящая только «хорошие» приближения, степени которых равны степеням двойки. Слабым звеном обоих подходов является необходимость использования плохо параллелизуемого алгоритма умножения матричных полиномов [146, 149], который, кроме этого, требует увеличения используемой оперативной памяти. Для простоты в дальнейшем будем полагать $n' = m' = n$.

Ещё одно важное замечание состоит в том, что во всех этих алгоритмах дважды последовательно строятся блоки $A^iY, i = 0, \ldots, \approx \frac{2N}{n}$, в то время как умножение на матрицу A - наиболее трудоёмкая операция.

Блочный фактор

Применение блочного фактора, то есть замены n на $n_1 = ns$, для алгоритмов типа Видемена-Копперсмита весьма эффективно, так как позволяет при помощи распараллеливания в s раз уменьшить время на производство шагов 2 и 4 (см. Приложение).

Важно отметить, что алгоритм умножения матричных полиномов, применявшийся при постановке последнего рекорда, потребовал дополнительного, по сравнению с оригинальной версией Копперсмита, объёма оперативной памяти (всего около 1ТВ при $s = 8$). Таким образом, ограничения оперативной памяти используемого кластера не позволили взять большее значение блочного фактора для уменьшения времени работы программы в целом.

Быстрые алгоритмы построения приближений

В последнее время появились несколько алгоритмов, которые строят приближения, удовлетворяющие нужным свойствам (например, невырождены

в некотором смысле), к формальным рядам рекурсивно [161, 148, 157]. Суть их в следующем: Пусть имеется приближение $G_1(\lambda) \in \mathbb{F}^{m\times m}[\lambda]$ порядка d к формальному ряду $\alpha(\lambda) \in \mathbb{F}^{l\times m}[[\lambda]]$:

$$\alpha(\lambda)G_1(\lambda) = O(\lambda^d).$$

Пусть также построено приближение G_2 порядка d к ряду $\frac{\alpha(\lambda)G_1(\lambda)}{\lambda^d}$ и оба эти приближения удовлетворяют некоторым нужным для алгоритма свойствам. Тогда, в ряде случаев, удаётся доказать, что произведение матриц $G_1 \cdot G_2$, являющееся, очевидно, приближением порядка $2d$ удовлетворяет тем же свойствам. Таким образом, задача построения приближения достаточно большого порядка 2^k сводится к построению двух приближений порядка 2^{k-1} и умножению матричных полиномов. Степени этих полиномов, в рассматриваемых раслучаях, не превосходят порядка соответствующего приближения и, в среднем по столбцам, равны его половине (см. например [146]). Отметим, что для вычисления матрицы G_2 необходимо, чтобы матрица G_1 уже была вычислена, поэтому данные вычисления не являются параллельными. Легко понять, что для вычисления приближения порядка 2^k этим алгоритмом нам придётся вычислить 2^i приближений порядка 2^{k-i}, $i = 0, 1, \ldots, k$. Общее число необходимых приближений становится в два раза больше их числа при последовательном построении, однако непосредственно строятся только 2^k приближений первой степени. Построение каждого из остальных сводится к одному умножению матричных полиномов, при этом, общее число операций в поле коэффициентов рассматриваемых многочленов оценивается величиной $O(2^k \cdot k^c)$, где c - некоторая небольшая абсолютная константа.

Однако для работы с матричными многочленами высокой степени требуется большая оперативная память, которая значительно возрастает из-за необходимости применения быстрых алгоритмов умножения матричных полиномов на одном кластере, что становится критическим для современной

техники. Этот дополнительный рост выражается в некотором логарифмическом множителе к объёму оперативной памяти, необходимой для хранения рассматриваемых многочленов. Этого нет при использовании последовательных алгоритмов. Сложность этих алгоритмов несколько выше и составляет $O(2^{2k})$, однако, как указал Копперсмит [146], это может оказаться несущественным, поскольку главный член в оценке времени связан не с этими вычислениями, а с построением самого ряда α. Как мы указывали выше, в алгоритме Видемана-Копперсмита приходится фактически дважды вычислять этот ряд. В данном разделе предложена экономия в этом месте за счёт указанного увеличения количества операций на использование последовательного алгоритма построения приближений.

Связь между приближениями рядов, зеркало

Пусть

$$\alpha(\lambda) = \sum_{i=0}^{\infty} \alpha_i \lambda^i \in \mathbb{F}^{n\times n}[[\lambda]].$$

Рассмотрим преобразование "точка"(иногда называют "зеркало"), которое действует на формальные ряды следующим образом:

$$\dot{\alpha}(\lambda) = \alpha(\lambda^{-1}),$$

а на многочлены фиксированной степени t так (см. также операцию rev в [146]):

$$\dot{Q} = \dot{Q}(\lambda) = \lambda^t Q(\lambda^{-1}).$$

Рассмотрим приближения типа Паде к ряду по положительным степеням:

$$\alpha(\lambda)Q^{(t)}(\lambda) + P^{(t)}(\lambda) = \sum_{i=2t+1}^{\infty} \rho_i^{(t)} \lambda^i,$$

$$Q^{(t)}(\lambda) \in \mathbb{F}^{n\times 2n}[\lambda], P^{(t)}(\lambda) \in \mathbb{F}^{n\times 2n}[\lambda], deg Q^{(t)}, P^{(t)} \leq t.$$

В некоторых работах ([148, 157]) приняты несколько другие обозначения:

$$(\alpha(\lambda)\|I_n)\begin{pmatrix} Q^{(t)} \\ P^{(t)} \end{pmatrix} = O(\lambda^{2t+1}) \tag{6.67}$$

(I_n - единичная матрица размера n), что будем обозначать как $ord^+G^{(t)} \geq 2t+1$, где

$$G^{(t)} = \begin{pmatrix} Q^{(t)} \\ P^{(t)} \end{pmatrix} \tag{6.68}$$

Матрицу $Q^{(t)}$ будем называть верхней частью $G^{(t)}$.

Нетрудно понять, что $\dot{Q}^{(t)}, \dot{P}^{(t)}$ будут матричными приближениями типа Паде к соответствующему ряду по отрицательным степеням переменной:

$$\dot{\alpha}(\lambda)\dot{Q}^{(t)}(\lambda) + \dot{P}^{(t)}(\lambda) = \sum_{i=t+1}^{\infty} \rho_i^{(t)} \lambda^{-i},$$

или

$$(\dot{\alpha}(\lambda)\|I_n)\begin{pmatrix} \dot{Q}^{(t)} \\ \dot{P}^{(t)} \end{pmatrix} = O(\lambda^{-(t+1)}), \tag{6.69}$$

что будем обозначать как $ord^-\dot{G}^{(t)} \geq t+1$, где

$$\dot{G}^{(t)} = \begin{pmatrix} \dot{Q}^{(t)} \\ \dot{P}^{(t)}. \end{pmatrix}. \tag{6.70}$$

По построению для произвольного матричного многочлена подходящего размера: $degG = deg\dot{G}$, $\dot{\dot{G}} = G$, $ord^+\dot{G} = ord^-G + degG$, или $ord^+\dot{G} - 2deg\dot{G} = ord^-G - degG = c(G)$. Значит, приближения Паде при преобразовании "точка"переходят в приближения Паде, а свойство $c(G) \geq 1$ является для них характеристическим.

Отметим здесь для дальнейшего, что если

$$\dot{\alpha}(\lambda)\tilde{Q}^{(t)}(\lambda)+\tilde{P}^{(t)}(\lambda)=\sum_{i=t+1-\delta}^{\infty}\tilde{\rho}_i^{(t)}\lambda^{-i},\delta\in\{0,1,\dots\,t\},deg\tilde{Q}^{(t)},\tilde{P}^{(t)}=t,$$

то

$$\alpha(\lambda)\dot{\tilde{Q}}^{(t)}(\lambda)+\dot{\tilde{P}}^{(t)}(\lambda)=\sum_{i=2t+1-\delta}^{\infty}\tilde{\rho}_i^{(t)}\lambda^{i},$$

или

$$ord^{+}\dot{\tilde{G}}^{(t)}=ord^{+}\begin{pmatrix}\dot{\tilde{Q}}^{(t)}\\ \dot{\tilde{P}}^{(t)}\end{pmatrix}\geq 2t+1-\delta \tag{6.71}$$

При этом, если при $2t\geq\delta$ свободный член многочлена $\dot{\tilde{Q}}^{(t)}$ равен нулю, то и свободный член многочлена $\dot{\tilde{P}}^{(t)}$ тоже равен нулю.

Таким образом, $ord^{+}\lambda^{\delta}\dot{\tilde{G}}^{(t)}(\lambda)\geq 2t+1$, и δ младших коэффициентов этого матричного многочлена равны нулю. Приближения, обладающие этими свойствами, очевидно, являются линейным подпространством над $\mathbb{F}[\lambda]$.

Как уже отмечалось, мы рассматриваем в данной работе только симметричный случай: $A=A^{T}$.

Определим для произвольного многочлена $Q(\lambda)=\sum_{i=0}^{degQ}\lambda^{i}Q_i\in\mathbb{F}^{n\times 2n}[\lambda]$, матрицы $A\in\mathbb{F}^{N\times N}$ и начального блока $B\in\mathbb{F}^{N\times 2n}$ блоки векторов $Q(A,B)^{-}$ и $Q(A,B)^{+}$ по следующим формулам:

$$Q(A,B)^{-}=\sum_{i=0}^{degQ}A^{i}BQ_i, \tag{6.72}$$

$$Q(A,B)^{+}=\sum_{i=0}^{degQ}A^{degQ-i}BQ_i=\dot{Q}(A,B)^{-}. \tag{6.73}$$

Отметим сразу, что если рассматривать многочлен Q, как многочлен степени $degQ+1$ с нулевым старшим коэффициентом, то определяемый блок получится

из предыдущего умножением слева на A, то есть данное определение зависит от формальной степени многочлена, под которой мы в дальнейшем, если это не ясно по умолчанию, будем понимать степень ненулевого монома максимальной степени данного многочлена, добавляя, в случае необходимости рассмотрения многочлена как многочлена большей степени, явно указанные нулевые мономы больших степеней. Формула (6.72) уже рассматривалась выше [72].

Важно отметить, что если $B \in \mathbb{F}^{N\times n}, Q^{(i)}(\lambda) \in \mathbb{F}^{n\times n}[\lambda], deg Q^{(i)}(\lambda) = i$, и свободные члены этих многочленов являются невырожденными матрицами, то имеет место равенство линейных пространств над $\mathbb{F}$:

$$\langle A^i B, i = 0, \ldots, t\rangle = \langle Q^{(i)}(A, B)^+, i = 0, \ldots, t\rangle.$$

Определим скалярное произведение $(Q_1, Q_2)^+$ матричных полиномов $Q_1(\lambda), Q_2(\lambda) \in \mathbb{F}^{n\times 2n}[\lambda]$ как коэффициент при $\lambda^{degQ_1+degQ_2+1}$ в ряде

$$Q_1^T(\lambda)\alpha(\lambda)Q_2(\lambda).$$

Легко видеть, что это скалярное произведение совпадает с коэффициентом при λ^{-1} в ряде

$$\dot{Q}_1^T(\lambda)\dot{\alpha}(\lambda)\dot{Q}_2(\lambda),$$

то есть со скалярным произведением $(\dot{Q}_1, \dot{Q}_2)^-$, определённым в [72] и линейным по обоим аргументам. Нетрудно понять, что преобразование "точка"сохраняет произведение:

$$(Q_1\dot{Q}_2) = \dot{Q}_1\dot{Q}_2,$$

и

$$(Q_1 \dot{+} Q_2) = \dot{Q}_1 \oplus \dot{Q}_2,$$

где $\oplus$ означает сложение полиномов путём сложения коэффициентов при старших степенях, затем при степенях на единицу меньших и так далее с присвоением формальной степени суммы максимума формальных степеней слагаемых. Поэтому, в силу (6.73) определённое нами $(Q_1, Q_2)^+$ будет линейным относительно операции $\oplus$. Поскольку, как было доказано в [72]

$$(\dot{Q}_1(\lambda), \dot{Q}_2(\lambda))^- = (\dot{Q}_1(A,B)^-)^T A \dot{Q}_2(A,B)^-,$$

то

$$(Q_1(\lambda), Q_2(\lambda))^+ = (\dot{Q}_1(\lambda), \dot{Q}_2(\lambda))^- = (\dot{Q}_1(A,B)^-)^T A \dot{Q}_2(A,B)^-$$
$$= (Q_1(A,B)^+)^T A Q_2(A,B)^+.$$

Последовательный алгоритм

Покажем, как при помощи скалярного произведения матричных полиномов можно последовательно строить решение разреженной симметричной системы линейных уравнений при использовании техники [146]. В работе [146] приближения $G^{(i)}(\lambda) \in \mathbb{F}^{2n\times 2n}[\lambda]$ ряда $(\alpha(\lambda) \parallel I_n) \in \mathbb{F}^{n\times 2n}[[\lambda]]$ строятся последовательно.

Столбцы матрицы

$$G^{(1)}(\lambda) = \begin{pmatrix} I_n & O_n \\ \alpha_0 & \lambda I_n \end{pmatrix}, \tag{6.74}$$

очевидно, образуют базис приближений порядка один. Пусть для некоторого i:

$$(\alpha(\lambda) \parallel I_n) G^{(i)}(\lambda) = C_i \lambda^i (1 + O(\lambda)), C_i \in \mathbb{F}^{n\times 2n}. \tag{6.75}$$

и столбцы матрицы $G^{(i)}(\lambda)$ образуют базис всех приближений порядка i. Выбирается невырожденная матрица $\tau_i \in \mathbb{F}^{2n\times 2n}$, при которой $C_i\tau_i$ имеет нижнетреугольный вид, а число нулевых столбцов в ней обозначим n_i. Тогда

$$G^{(i+1)}(\lambda) = G^{(i)}(\lambda)\tau_i \begin{pmatrix} \lambda I_{2n-n_i} & O \\ O & I_{n_i} \end{pmatrix}. \tag{6.76}$$

Поскольку линейное пространство приближений порядка $i+1$ вложено в линейное пространство приближений порядка i, порождённое столбцами $G^{(i)}(\lambda)\tau_i$, то из (6.75) следует, что столбцы $G^{(i+1)}(\lambda)$ образуют базис приближений порядка $i+1$.

Если обозначить

$$G^{(i)}(\lambda) = \begin{pmatrix} Q^{(i)}(\lambda) \\ P^{(i)}(\lambda) \end{pmatrix}, Q^{(i)}(\lambda), P^{(i)}(\lambda) \in \mathbb{F}^{n\times 2n}[\lambda], \tag{6.77}$$

то по построению $degQ^{(i)}(\lambda) \leq i-1$ и

$$Q^{(i+1)}(\lambda) = \lambda Q^{(i)}(\lambda)\tau_{i1} + Q^{(i)}(\lambda)\tau_{i2} = \lambda Q^{(i)}(\lambda)\tau_{i1} \oplus (Q^{(i)}(\lambda)\tau_{i2} + O_{2n}\lambda^{degQ^{(i)}+1}),$$

для некоторых матриц $\tau_{i1}, \tau_{i2} \in \mathbb{F}^{2n\times 2n}$, где $O_{2n} \in \mathbb{F}^{n\times 2n}$ - нулевая матрица. Поэтому

$$Q^{(i+1)}(A,B)^+ = Q^{(i)}(A,B)^+\tau_{i1} + AQ^{(i)}(A,B)^+\tau_{i2}.$$

Поскольку $degQ^{(i)}(\lambda) = i-1$, то скалярное произведение

$$(Q^{(1)}(A,B)^+)^T AQ^{(i+1)}(A,B)^+ = (Q^{(1)}(\lambda), Q^{(i+1)}(\lambda))^+ = {Q_0^{(1)}}^T C_{i+1} = \begin{pmatrix} C_{i+1} \\ O_n \end{pmatrix} \tag{6.78}$$

с учётом предыдущего равенства позволяет вычислить сам коэффициент C_{i+1}, а также

$$Q^{(i+1)}(A,B)^+, AQ^{(i+1)}(A,B)^+, \ldots, A^{k-1}Q^{(i+1)}(A,B)^+$$

непосредственно из блоков

$$Q^{(i)}(A,B)^+, AQ^{(i)}(A,B)^+, A^2Q^{(i)}(A,B)^+, \ldots, A^kQ^{(i)}(A,B)^+,$$

не используя коэффициенты ряда $\alpha(\lambda)$, которые в данной версии вычислять не нужно. Таким образом в шаге алгоритма используется только одно умножение на матрицу A блока $A^{k-1}Q^{(i+1)}(A,B)^+ \in \mathbb{F}^{N\times 2n}$ для получения очередной последовательности блоков $Q^{(i+1)}(A,B)^+, AQ^{(i+1)}(A,B)^+, \ldots, A^kQ^{(i+1)}(A,B)^+$.

Для построения всего пространства Крылова в виде описанных выше блоков нам необходимо построить последовательность приближений

$$\tilde{\tilde{G}}^{(r)}(\lambda) = \begin{pmatrix} \tilde{\tilde{Q}}^{(r)}(\lambda) \\ \tilde{\tilde{P}}^{(r)}(\lambda) \end{pmatrix} \in \mathbb{F}^{2n\times n}[\lambda], deg\tilde{\tilde{G}}^{(r)} = r, r = 0, 1, \ldots, \approx \frac{N}{n}, \qquad (6.79)$$

верхняя часть свободных членов которых представляет невырожденные матрицы. Для этого, будем использовать приближения $G^{(2r+1)}(\lambda)$, построенные в шагах с нечётными номерами, с которыми будем делать следующие преобразования:

Пусть $d(j, 2r+1)$ — степень j-ого столбца $G^{(2r+1)}(\lambda)$ как матричного многочлена.

Итерация 1: Рассмотрим столбцы, степени которых $d(j, 2r+1) \leq r$. Если верхние части их свободных членов образуют матрицу ранга n, то $\tilde{\tilde{G}}^{(r)}(\lambda)$ можно составить из тех из них, верхние части свободных членов которых линейно независимы.

Итерация 2: В противном случае, рассмотрим линейное подпространство над $\mathbb{F}[\lambda]$, относительно обычного сложения, столбцов с $d(j, 2r+1) \leq r+1$, верхние части свободных членов которых нулевые. Поскольку эти столбцы являются приближениями, их свободные члены полностью нулевые. В это пространство войдут умноженные на λ столбцы, выбранные на первом шаге. Если они не являются базисом всего рассматриваемого подпространства, выберем в нём столбцы, образующие базис дополнения. Разделив эти столбцы на λ, получим столбцы степени r, порядка $2r$. Дополним ими столбцы, выбранные на 1 шаге. Если число выбранных столбцов равно n, мы нашли $\tilde{\tilde{G}}^{(r)}(\lambda)$.

Итерация 3: Если на втором шаге набрать нужные столбцы не удалось,

аналогично рассматриваем столбцы с условием $d(j,2r+1) \le r+2$ и верхними частями двух младших коэффициентов, равными нулю, и т.д.

Обозначим число необходимых итераций $v(r)$ и рассмотрим его как случайную величину, зависящую от матричных коэффициентов используемых здесь приближений, которые будем считать независимыми случайными величинами.

Теорема 6.9. *Матожидание* $v(r) \le 1,76$. *При* $\log_2 \frac{N}{n} > 10$ *с вероятностью 0,99* $\max_s v(s) \le 4 \log_2 \frac{N}{n}$.

Доказательство. В работе [72] было показано, как построить приближения $\tilde{Q}^{(s)}(\lambda) \in \mathbb{F}^{n\times n}[\lambda], deg\tilde{Q}^{(s)} = s$ с невырожденными старшими коэффициентами $\tilde{Q}_s^{(s)}$ к ряду $\dot{\alpha}(\lambda) = \sum_{i=0}^{\infty} \alpha_i \lambda^{-i}$ так, чтобы

$$\dot{\alpha}(\lambda)\tilde{Q}^{(s)}(\lambda) - \tilde{P}^{(s)}(\lambda) = \sum_{i=s+1-\delta(s)}^{\infty} \tilde{\alpha}_i^{(s)} \lambda^{-i}, \tag{6.80}$$

для некоторого $\tilde{P}^{(s)}(\lambda) \in \mathbb{F}^{n\times n}[\lambda], deg\tilde{P}^{(s)} = s$. При небольших значениях $\delta(s)$. При этом ненулевые столбцы в матрицах $\tilde{\alpha}_{s+1-l}^{(s)}, l > 0$ расположены справа, а их число $u^{(s)}(l)$ не возрастает с ростом l.

Имеем

$$\alpha(\lambda)\dot{\tilde{Q}}^{(s)}(\lambda) - \dot{\tilde{P}}^{(s)}(\lambda) \;=\; \sum_{i=2s+1-\delta(s)}^{\infty} \tilde{\alpha}_i^{(s)} \lambda^{i}. \tag{6.81}$$

Обозначим

$$\dot{\tilde{G}}^{(r)} = \begin{pmatrix} \dot{\tilde{Q}}^{(r)} \\ \dot{\tilde{P}}^{(r)} \end{pmatrix}. \tag{6.82}$$

Этот матричный многочлен имеет невырожденную верхнюю часть свободного члена и может быть взят в качестве $\tilde{\tilde{G}}^{(r)}$. Однако шаги 1-3 непосредственно строят подобный многочлен с заведомо не большим значением $\delta(r)$.

Столбцы многочленов $\lambda\dot{\tilde{G}}^{(r)}(\lambda)$, находящиеся на позициях

$$u^{(r)}(2)+1,\ldots,u^{(r)}(1), \tag{6.83}$$

считая справа, будут лежать в линейных пространствах, описанных в шаге 2. Поэтому, в случае $\delta(r)=1$, имеем: $u^{(r)}(2)=0$, и верхние части столбцов свободного члена многочлена $\dot{\tilde{G}}^{(r)}(\lambda)$ на позициях (6.83) дополняют верхние части свободных членов столбцов, выбранных на 1 шаге до матрицы полного ранга n, так как линейное пространство над $\mathbb{F}$ столбцов, выбранных на первом шаге в этом случае, содержит столбцы многочлена $\dot{\tilde{G}}^{(r)}(\lambda)$, находящиеся на позициях $u^{(r)}(1)+1,\ldots,n$, считая справа.

Соответственно столбцы, находящиеся на позициях $u^{(r)}(3)+1,\ldots,u^{(r)}(2)$, в многочлене $\lambda^2\dot{\tilde{G}}^{(r)}(\lambda)$ находятся в линейном пространстве, описанном в шаге 3, и так далее. Как было отмечено выше, алгоритм Видемана-Копперсмита строит матричные приближения, столбцы которых являются базисом над $\mathbb{F}[\lambda]$, относительно обычного сложения, всех приближений данного порядка. Поэтому рассматриваемые образы могут быть выбраны в шагах 2 и 3. Тем самым, число итераций $v(r)$, подобных шагам 2 и 3 для построения приближения с невырожденным свободным членом, не превосходит $\delta(r)$.

Как было вычислено в [152], при $\log_2\frac{N}{n}>10$ с вероятностью 0,99, $\max_s\delta(s)\le 4\log_2\frac{N}{n}$, в то время как среднее значение $\delta(s)$ не превосходит 1,76.

Теорема доказана.

Общее число шагов построения приближений в нашем алгоритме примерно равно $\frac{2N}{n}$. Для построения пространства Крылова будет использована половина из них.

Построенные $\tilde{\tilde{Q}}^{(i)}(\lambda)$ имеют степень i и невырожденные свободные члены. Порядок их будет меньше, чем $2i+1$ на число $\delta'(i)\le\delta(i)$:

$$\alpha(\lambda)\tilde{\tilde{Q}}^{(i)}(\lambda) - \tilde{\tilde{P}}^{(i)}(\lambda) \;=\; \sum_{l=2i+1-\delta'(l)}^{\infty} \tilde{\tilde{\alpha}}_l^{(i)} \lambda^l, \tag{6.84}$$

Поэтому, при $j < i - \delta'(i)$,

$$(\tilde{\tilde{Q}}^{(j)}(A,B)^+)^T A \tilde{\tilde{Q}}^{(i)}(A,B)^+ = 0,$$

а при $i - \delta'(i) \le j \le i$:

$$(\tilde{\tilde{Q}}^{(j)}(A,B)^+)^T A \tilde{\tilde{Q}}^{(i)}(A,B)^+ = (\tilde{\tilde{Q}}^{(j)}(\lambda), \tilde{\tilde{Q}}^{(i)}(\lambda))^+ = \sum_{k+l=j+i+1} \tilde{\tilde{Q}}_k^{(j)T} \tilde{\tilde{\alpha}}_l^{(i)}, \tag{6.85}$$

где $\tilde{\tilde{Q}}^{(j)}(\lambda) = \sum_{l=0}^{j} \tilde{\tilde{Q}}_l^{(j)} \lambda^l$, и сумма справа в (6.85) содержит не более $\delta'(l)+1$ слагаемых.

По построению $\tilde{\tilde{Q}}^{(r)}(\lambda) = Q^{(2r+1)}(\lambda)(\tau_{r0} + \frac{1}{\lambda}\tau_{r1} + \cdots + \frac{1}{\lambda^{\delta'(r)}}\tau_{r\delta'(r)})$ для некоторых $\tau_{ri} \in \mathbb{F}^{2n\times 2n}$. Поэтому

$$\tilde{\tilde{Q}}^{(r)}(A,B)^+ = \sum_{i=0}^{\delta'(r)} A^i Q^{(2r+1)}(A,B)^+ \tau_{ri},$$

и при $k = \max_i \delta'(i)$ мы, используя имеющуюся на каждом шаге последовательность $Q^{(2r+1)}(A,B)^+, AQ^{(2r+1)}(A,B)^+, \ldots, A^k Q^{(2r+1)}(A,B)^+$, можем вычислить $\tilde{\tilde{Q}}^{(r)}(A,B)^+$.

Несложная доортогонализация приводит к последовательному построению A ортогонального базиса W_i пространства Крылова одновременно с очередным слагаемым в сумме:

$$X = \sum_{i=0}^{\approx \frac{N}{n}} W_i (W_i^T A W_i)^{-1} W_i^T B, \tag{6.86}$$

при помощи которого решение системы строится также, как в алгоритме Монтгомери [145].

Доортогонализация требует временного хранения последовательных блоков пространства Крылова, не ортогональных друг другу, и вычисления их скалярных произведений. Число этих блоков определяется величиной $\delta'(s) \leq \delta(s)$. Если ограничиться хранением в оперативной памяти лишь $k \approx 7$ последовательных блоков пространства Крылова, а предыдущие $O(lnN)$ хранить в долговременной памяти на жёстком диске, то, ввиду редкого $\left(\approx \frac{N}{2k}\right)$ к ней обращения, оно не повлияет на скорость работы всего алгоритма.

В соответствии с результатами численного эксперимента, есть все основания полагать, что реальное время работы нашего алгоритма значительно меньше: шаг 3 не нужен, а иногда не нужен и шаг 2.

Грубая оценка снизу требуемого объёма используемой оперативной памяти даёт

$$\max_s \delta(s) \cdot Nn = O(nNlnN),$$

что по порядку совпадает с требованииями для алгоритма Томэ [161]. Однако, если $\delta(s) \in \{0, 1\}$, эти требования существенно снижаются.

Параллельный алгоритм

Коэффициенты C_i в шаге алгоритма можно строить, как в [146], при помощи свёртки полинома $G^{(i)}$ и ряда α. Ввиду формул (6.85), доортогонализацию тоже можно проводить для многочленов $\tilde{\tilde{Q}}^{(i)}(\lambda)$, а $\tilde{\tilde{\alpha}}_l^{(i)}(\lambda)$ строить при помощи свёрток с рядом α. В результате, будем получать ортогональный базис пространства Крылова W_i в виде многочленов $W_i(\lambda)$, для которых $W_i = W_i(A, B)^+$. Поэтому формула (6.86) даст $X(\lambda), degX \approx \frac{N}{n}$, такой, что $X = X(A, B)^+$, который вычисляется в результате повторного параллельного вычисления с использованием блочного фактора элементов $A^i B, i = 0, 1, \ldots, \approx \frac{N}{n}$.

Комментарии

Предложенный в этом параграфе последовательный алгоритм, по существу, выполняет те же действия, что и алгоритм Монтгомери [145]. Количество

умножений на матрицу A в нём вдвое больше. Применение техники построения приближений [146] позволяет уменьшить время на доортогонализацию получаемых блоков относительно уже построенных, сократив их число с 3 до не более, чем одного. Применение параллельных систем в этом случае сковано необходимостью собирать на каждом шаге матрицы C_i (6.78).

Описанный параллельный алгоритм работает по схеме алгоритма Видемана-Копперсмита и позволяет уменьшить число умножений на матрицу A при построении решения в последнем шаге (исключить вычисление f в шаге 4 (см. Приложение)) за счёт некоторого увеличения числа операций на построение его координат. $\frac{N}{n}$ двойных умножений на A и на A^T, как мы видели выше, будет проще $\frac{2N}{n}$ однократных умножений на A, а решение будет линейной комбинацией $A^iB, i = 0, 1, \ldots, \approx \frac{N}{n}$. Кроме того, если строить последовательные приближения кусками с вычислением части координат решения, то можно совсем отказаться от 3 и 4 шагов, запоминая только кусок пространства Крылова. В этом случае не будет необходимости иметь большую память и можно отказаться от использования кластеров большой мощности в решении рассматриваемой задачи.

Если сравнивать трудоёмксть этого алгоритма с трудоёмкостью алгоритма Монтгомери, то следует отметить, что левый множитель в формуле для вычисления скалярных произведений (6.78) не меняется, и, значит, может храниться на вычислительном узле, вычисляющем правый множитель. Тем самым, экономится время на пересылки при вычислении скалярных произведений, которое имеет доминирующую роль. Скалярное произведение векторов заменено на скалярное произведение многочленов, которое проще и требует меньше оперативной памяти. Количество сложений векторов в каждых двух шагах предложенного алгоритма примерно в два раза меньше, чем в одном шаге алгоритма Монтгомери, имеющего в два раза меньше шагов. Число векторов, которые надо хранить, примерно в четыре раза меньше. Аналогичные построения могут быть применены и для алгоритма построения σ-базиса

[157]. Конечно, рассматриваемый алгоритм использует вдвое большее число умножений на большую разреженную матрицу, чем алгоритм Монтгомери. Однако, эти умножения используются в процедуре последовательного вычисленя коэффициентов ряда, которая легко параллелизуется на вычислительные ресурсы, не связанные быстрыми каналами, то есть могут быть проделаны в интернете. Это удешевляет вычисления. Видимо поэтому математиками, поставившими рекорд целой факторизации 12 декабря 2009 года был выбран алгоритм Видемана-Копперсмита, а не алгоритм Монтгомери.

В перспективе этот алгоритм можно дополнить процедурой синхронизации, что даст возможность использовать ещё один параметр для настройки программы под конкретную вычислительную сеть.

Глава 7

Решение разреженных систем линейных уравнений над $GF(p)$

В этом разделе предложен универсальный алгоритм, предназначенный для решения больших разреженных систем линейных уравнений над конечными полями с большим простым числом элементов, которые возникают при решении задачи дискретного логарифмирования по модулю простого числа. Алгоритм разработан для использования на параллельных вычислительных системах с разнообразной параллельной архитектурой и характеристиками. Новый метод наследует структурные свойства метода Ланцоша, позволяя, однако, гибко управлять сложностью параллельных вычислений и количеством обменов.

В настоящее время среди методов, которые используются для решения больших разреженных систем над конечными полями, одним из наиболее часто упоминаемых является метод Ланцоша (см., например [168], [171], [169], [170], [172], [173], [174], [72], [175], [176]). Простота алгоритма – главное достоинство метода. Однако по своей природе метод Ланцоша является последовательными алгоритмом, на каждой итерации которого происходит большой обмен данными между различными вычислительными узлами.

Несмотря на то, что использование блочных версий алгоритма Ланцоша несколько улучшает масштабируемость параллельных программ (см. например,

[176]), но так или иначе не позволяет эффективно использовать вычислительные системы с производительностью более 100 TFlops, причем недостатки всех программных реализаций метода Ланцоша становятся тем более заметными, чем ниже скорость обмена данными в коммуникационной сети вычислительной системы.

В настоящей работе предлагается подход к построению *универсального* блочного метода Ланцоша-Паде для решения больших разреженных систем линейных уравнений над конечными полями с большим (на практике порядка 10^{150}) числом элементов. Термин «универсальный» используется в работе как замена выражению «эффективный на параллельных вычислительных системах с различной архитектурой и характеристиками». В предлагаемом подходе имеется возможность гибко управлять сложностью вычислений и размером обмениваемых данных. При построении универсального метода Ланцоша-Паде мы следуем работам [167], [72], [77].

7.1 A-ортогональный базис пространства Крылова и метод Ланцоша

Пусть $A \in \mathbb{F}^{n \times n}$ – симметричная матрица, а $B \in \mathbb{F}^{n \times K}$ – блок, составленный из K векторов, с элементами из «большого» конечного простого поля $\mathbb{F}$. Рассмотрим систему линейных уравнений с матрицей A и правой частью B

$$AX = B. \tag{7.1}$$

Для решения системы (7.1) можно использовать блочный метод Ланцоша, $k+1$-ая итерация которого состоит из следующих операций:

1. вычислить произведение AQ_k матрицы A на блок $Q_k \in \mathbb{F}^{n \times K}$
2. построить новый блок направлений $Q_{k+1} \in \mathbb{F}^{n \times K}$ при помощи коротких

рекуррентных соотношений:

$$Q_{k+1} \;=\; AQ_k - Q_k C_k - Q_{k-1} C_{k-1}, \tag{7.2}$$

с матрицами коэффициентов C_k и C_{k-1} размера $K \times K$, вычисляемыми по следующим формулам:

$$C_k = \left(Q_k^T A Q_k\right)^{-1} Q_k^T A^2 Q_k,\; C_{k-1} = \left(Q_{k-1}^T A Q_{k-1}\right)^{-1} Q_{k-1}^T A^2 Q_k.$$

Замечание. *Для блоков направлений Q_s справедливы соотношения A-ортогональности (то есть $Q_l^T A Q_r = 0$ для $l \neq r$) [171]. Кроме того, в дальнейшем будем предполагать, что все матрицы вида $Q_s^T A Q_s, s \in \{1, \ldots, \frac{n}{K} - 1\}$ являются невырожденными. Это справедливо с вероятностью близкой к 1 в предположении, что матрицы $Q_s^T A Q_s$ являются независимыми случайными симметричными матрицами с элементами из поля $\mathbb{F}$. Справедливость такого предположения для задачи дискретного логарифмирования вытекает из того, что размер исходной матрицы существенно меньше характеристики поля, и подтверждается многочисленными численными экспериментами (см. например, [173], [174])*;

3. вычисление нового «приближенного» решения X_{k+1} в виде

$$\begin{aligned} X_{k+1} \;&=\; \sum_{i=1}^{k+1} Q_i \left(Q_i^T A Q_i\right)^{-1} Q_i^T B \\ &=\; X_k + Q_{k+1} \left(Q_{k+1}^T A Q_{k+1}\right)^{-1} Q_{k+1}^T B. \end{aligned}$$

Из сделанного предположения, что матрицы $Q_s^T A Q_s$ невырожденные, следует, что каждый блок направлений Q_s имеет полный ранг K (т.е., что столбцы, образующие Q_s, являются линейно независимыми), а линейная оболочка столбцов, из которых составлены блоки Q_1, Q_2, $\cdots$, Q_s, совпадает с линейной болочкой столбцов, входящих в блоки B, AB, $\cdots$, $A^{s-1}B$.

В случае конечных полей метод Ланцоша является итерационной процедурой лишь условно. Для того чтобы найти решение задачи (7.1) требуется построить пространство Крылова (блочное пространство Крылова) полной размерности. Таким образом, метод Ланцоша – это существенно последовательная процедура построения A-ортогонального базиса $Q_1\,,\ Q_2\,,\ \cdots\,,\ Q_{n/K}$ пространства Крылова $\mathcal{K}(A,B)$

$$\mathcal{K}(A,B) = \ \mathrm{Span}\{B, AB, A^2B, \cdots, A^{\frac{n}{K}-1}B\}$$

и одновременного вычисления коэффициентов разложения решения X в построенном базисе. Каждый новый блок направлений Q_{s+1} может быть вычислен только при условии, что известны предыдущие блоки $Q_1\,,\ Q_2\,,\ ...,\ Q_s\,.$

Последовательная структура алгоритма Ланцоша подразумевает, что ресурс независимых вычислений, необходимый при реализации метода на параллельных вычислительных системах, весьма ограничен. Более того, явное (непосредственное) вычисление векторов направлений Q_s, как правило (см. например, [176]), сопровождается существенным обменом данными между различными вычислительными узлами, что увеличивает время работы алгоритма на вычислительных системах с медленными межузловыми связями.

Идея универсального алгоритма состоит в том, чтобы не вычислять блоки Q_s явно, а использовать отрезки степенного базиса пространства Крылова. Относительно большой обмен данными в универсальном алгоритме будет иметь место только в некоторые моменты инициализации новых отрезков степенного базиса. Частота таких инициализаций является параметром алгоритма. Именно возможность управлять размером обмениваемых данных является главной особенностью нового алгоритма.

7.2 A-ортогональный базис пространства Крылова и матричные приближения Паде

Нам потребуется ряд конструкций, чтобы установить связь между матричными приближениями Паде некоторого специального ряда и A-ортогональными базисами пространства Крылова $\mathcal{K}(A,B)$.

Итак, пусть $\alpha(x)$ – формальный матричный ряд

$$\alpha(x) = \sum_{i=0}^{\infty} \alpha_i x^{-i}, \tag{7.3}$$

коэффициентами которого являются матрицы $\alpha_i \in \mathbb{F}^{K\times K}$. Будем говорить, что пара матричных полиномов $\mathcal{Q}^{(s)}(x)$ и $\mathcal{P}^{(s)}(x)$ степени s

$$\begin{aligned}\mathcal{Q}^{(s)}(x) &= \mathcal{Q}_0^{(s)} + \mathcal{Q}_1^{(s)}x + \cdots + \mathcal{Q}_s^{(s)}x^s,\\ \mathcal{P}^{(s)}(x) &= \mathcal{P}_0^{(s)} + \mathcal{P}_1^{(s)}x + \cdots + \mathcal{P}_s^{(s)}x^s,\end{aligned}$$

с коэффициентами $\mathcal{Q}_i^{(s)}, \mathcal{P}_j^{(s)} \in \mathbb{F}^{K\times K}$, задает *матричное приближение Паде степени* s для ряда (7.3), если выполняется соотношение:

$$\alpha(x)\mathcal{Q}^{(s)}(x) - \mathcal{P}^{(s)}(x) = \sum_{i=s+1}^{\infty} \rho_i^{(s)} x^{-i}. \tag{7.4}$$

Ряд в правой части (7.4) будем называть *остаточным рядом* приближения Паде $\mathcal{P}^{(s)}(x)$, $\mathcal{Q}^{(s)}(x)$ степени s и обозначать его через $\mathcal{R}^{(s)}(x)$.

Далее, для ряда (7.3) и пары произвольных матричных полиномов определим *билинейное отображение*. А именно, пусть $\mathcal{Q}_1(x)$ и $\mathcal{Q}_2(x)$ два матричных полинома. Рассмотрим ряд, полученный формальным произведением $\mathcal{Q}_1^T(x)$, $\alpha(x)$ и $\mathcal{Q}_2(x)$,

$$\mathcal{Q}_1^T(x)\alpha(x)\mathcal{Q}_2(x) = \sum_{i=-t}^{\infty} \gamma_i x^{-i}. \tag{7.5}$$

Тогда значение билинейного отображения $< \mathcal{Q}_1, \mathcal{Q}_2 > \in \mathbb{F}^{K\times K}$ для пары полиномов $\mathcal{Q}_1$ и $\mathcal{Q}_2$ определяется как значение коэффициента γ_1 формального

ряда (7.5):

$$< \mathcal{Q}_1, \mathcal{Q}_2 >= \gamma_1. \tag{7.6}$$

Наконец, установим способ построения блока Q размера $n \times K$ по заданному матричному полиному $\mathcal{Q}(x)$, квадратной $n \times n$ матрице A и $n \times K$ блоку B. Для этого рассмотрим полином $\mathcal{Q}(x)$ степени s с матричными коэффициентами $\mathcal{Q}_0$, $\mathcal{Q}_1$, $\cdots$, $\mathcal{Q}_s$:

$$\mathcal{Q}(x) = \mathcal{Q}_0 + \mathcal{Q}_1 x + \cdots + \mathcal{Q}_s x^s.$$

Определим блок $Q = \mathcal{Q}(A, B) \in \mathbb{F}^{n \times K}$ по следующему правилу (см. например, [72]):

$$Q = \mathcal{Q}(A, B) = B\mathcal{Q}_0 + AB\mathcal{Q}_1 + \cdots + A^s B\mathcal{Q}_s. \tag{7.7}$$

В дальнейшем нас будут интересовать только те ряды $\alpha(x)$ вида (7.3), коэффициенты α_i которых задаются некоторым специальным образом, а именно

$$\alpha_i = \alpha_i^T = B^T A^i B,$$

с симметричной матрицей A, и некоторым блоком B. То есть, начиная с этого момента и всюду далее, будем считать, что введенный нами ранее ряд $\alpha(x)$ имеет вид

$$\alpha(x) = \sum_{i=0}^{\infty} \left(B^T A^i B\right) x^{-i}. \tag{7.8}$$

Следущая лемма связывает значение билинейного отображения $< \mathcal{Q}_1, \mathcal{Q}_2 >$ для матричных полиномов $\mathcal{Q}_1$ и $\mathcal{Q}_2$ с матричным произведением блоков $Q_1 = \mathcal{Q}_1(A, B)$ и $Q_2 = \mathcal{Q}_2(A, B)$, порожденных для теми же самыми полиномами.

Лемма 1. [72] *Справедливо следующее равенство:*

$$< \mathcal{Q}_1, \mathcal{Q}_2 >= Q_1^T A Q_2 \tag{7.9}$$

С помощью леммы 1 докажем утверждение, которое связывает матричные приближения Паде различных степеней с некоторым A-ортогональным базисом пространства $\mathcal{K}(A, B)$.

Утверждение 1. *Пусть* $A \in \mathbb{F}^{n\times n}$ *симметричная матрица,* $B \in \mathbb{F}^{n\times K}$ *блок из* K *линейно независимых векторов, а* $\alpha_i \in \mathbb{F}^{K\times K}$ *квадратные матрицы вида* $\alpha_i = \alpha_i^T = B^T A^i B$. *Рассмотрим приближения Паде* $\mathcal{P}^{(s)}(x)$, $\mathcal{Q}^{(s)}(x)$ *ряда (7.8) степеней* $0, \cdots, l$,

$$\begin{aligned}
\alpha(x)\mathcal{Q}^{(0)}(x) - \mathcal{P}^{(0)}(x) &= \sum_{i=1}^{\infty} \rho_i^{(0)} x^{-i}, \\
\alpha(x)\mathcal{Q}^{(1)}(x) - \mathcal{P}^{(1)}(x) &= \sum_{i=2}^{\infty} \rho_i^{(1)} x^{-i}, \\
&\cdots \quad \cdots \quad \cdots, \\
\alpha(x)\mathcal{Q}^{(l)}(x) - \mathcal{P}^{(l)}(x) &= \sum_{i=l+1}^{\infty} \rho_i^{(l)} x^{-i},
\end{aligned}$$

предполагая, что матрицы $Q^{(s)}(A,B)^T A Q^{(s)}(A,B)$ *являются невырожденными.*

Тогда, для любого s, $0 \le s \le l$ *блоки* $Q_0 = \mathcal{Q}^{(0)}(A,B), \cdots, Q_l = \mathcal{Q}^{(l)}(A,B)$, *построенные по* $\mathcal{Q}$*-полиномам приближений Паде, задают* A*-ортогональный базис пространства Крылова* $\mathcal{K}_l(A,B)$

$$\mathcal{K}_l(A,B) = \mathrm{Span}\left(B, AB, \cdots, A^l B\right).$$

Кроме того, $Q^{(s)}(A,B)^T A Q^{(s)}(A,B) = Q_s^{(s)T} \rho_{s+1}^{(s)}$, *и старшие коэффициенты матричных* $\mathcal{Q}$*-полиномов и старшие коэффициенты соответствующих остаточных рядов, стоящие в последнем равенстве справа, являются невырожденными матрицами.*

Доказательство. *Рассмотрим два* $\mathcal{Q}$ *полинома* $\mathcal{Q}^{(m)}(x)$ *и* $\mathcal{Q}^{(k)}(x)$ *степени* m *и* k, *соответственно, причем* $m \neq k$. *Не теряя в общности, считаем, что* $m > k$. *Рассмотрим остаточный ряд* $\mathcal{R}^{(m)}(x)$ *для приближения* $\mathcal{Q}^{(m)}(x), \mathcal{P}^{(m)}(x)$ *степени* m

$$\alpha(x)Q^{(m)}(x) - P^{(m)}(x) = \sum_{i=m+1}^{\infty} \rho_i^{(m)} x^{-i}. \tag{7.10}$$

Старший ненулевой член ряда (7.10) имеет степень не выше $-(m+1)$, *следовательно, старший член произведения* $\mathcal{R}^{(m)}(x)$ *на матричный полином* $(\mathcal{Q}^{(k)}(x))^T$ *будет иметь степень не выше* $k-m-1$. *Принимая во внимание, что* $k<m$, *а, следовательно,* $k-m-1<-1$, *заключаем, что* $<\mathcal{Q}^{(m)},\mathcal{Q}^{(k)}>=0$. *Теперь непосредственно из леммы 1 вытекает* A*-ортогональность блоков* $\mathcal{Q}^{(m)}(A,B)$ *и* $\mathcal{Q}^{(k)}(A,B)$. *Осталось заметить, что размерности векторных пространств, образованных векторами в блоках* $\mathcal{Q}^{(0)}(A,B)$, $\cdots$, $\mathcal{Q}^{(l)}(A,B)$ *и* B, AB, $\cdots$, A^lB, *будут совпадать, если только старшие коэффициенты* $\mathcal{Q}$*-полиномов приближений Паде являются невырожденными.*

Остальное доказывается аналогично. $\square$

Утверждение 1 предоставляет иной ("неявный") способ описания A *– ортогональных базисов пространства Крылова* $\mathcal{K}(A,B)$. *А именно, чтобы задать* A*-ортогональный базис, достаточно знать соответствующие* $\mathcal{Q}$ *– полиномы приближений Паде. Вопросы вычисления* $\mathcal{Q}$*-полиномов с помощью коротких рекуррентных соотношений рассматриваются в следующем разделе 7.3.*

7.3 Рекуррентные формулы для приближений Паде

Предположим, что уже известны приближения Паде $\mathcal{P}^{(s-1)}$, $\mathcal{Q}^{(s-1)}$ *и* $\mathcal{P}^{(s)}$, $\mathcal{Q}^{(s)}$ *порядков* $s-1$ *и* s, *соответственно, с остаточными рядами* $\mathcal{R}^{(s-1)}(x)$ *и* $\mathcal{R}^{(s)}(x)$

$$\begin{aligned}\mathcal{R}^{(s-1)}(x) &= \alpha(x)\mathcal{Q}^{(s-1)}(x)-\mathcal{P}^{(s-1)}(x)=\sum_{i=s}^{\infty}\rho_i^{(s-1)}x^{-i},\\ \mathcal{R}^{(s)}(x) &= \alpha(x)\mathcal{Q}^{(s)}(x)-\mathcal{P}^{(s)}(x)=\sum_{i=s+1}^{\infty}\rho_i^{(s)}x^{-i}.\end{aligned}$$

Новое приближение Паде $\mathcal{Q}^{(s+1)}(x)$, $\mathcal{P}^{(s+1)}(x)$ *степени* $s+1$ *будем искать в виде*

$$\mathcal{Q}^{(s+1)}(x) = x\mathcal{Q}^{(s)}(x) + \mathcal{Q}^{(s)}(x)\nu_0 + \mathcal{Q}^{(s-1)}(x)\nu_1, \tag{7.11}$$

$$\mathcal{P}^{(s+1)}(x) = x\mathcal{P}^{(s)}(x) + \mathcal{P}^{(s)}(x)\nu_0 + \mathcal{P}^{(s-1)}(x)\nu_1, \tag{7.12}$$

$$\mathcal{R}^{(s+1)}(x) = x\mathcal{R}^{(s)}(x) + \mathcal{R}^{(s)}(x)\nu_0 + \mathcal{R}^{(s-1)}(x)\nu_1, \tag{7.13}$$

где ν_0 *и* ν_1 *–* $K \times K$ *матрицы, требующие определения. Из условия равенства нулю коэффициентов* $\rho_s^{(s+1)}$ *и* $\rho_{s+1}^{(s+1)}$ *остатка ряда Паде* $\mathcal{R}^{(s+1)}(x)$, *следует, что матрицы* ν_0 *и* ν_1 *удовлетворяют системе линейных уравнений*

$$\rho_{s+1}^{(s)} + \rho_s^{(s-1)}\nu_1 = \rho_s^{(s+1)} = 0, \tag{7.14}$$

$$\rho_{s+2}^{(s)} + \rho_{s+1}^{(s)}\nu_0 + \rho_{s+1}^{(s-1)}\nu_1 = \rho_{s+1}^{(s+1)} = 0. \tag{7.15}$$

Оставшиеся коэффициенты $\rho_{s+k+1}^{(s+1)}$ *остатка* $\mathcal{R}^{(s+1)}(x)$ *с* $k > 0$ *получаются по следующей формуле:*

$$\rho_{s+k+1}^{(s+1)} = \rho_{s+k+2}^{(s)} + \rho_{s+k+1}^{(s)}\nu_0 + \rho_{s+k+1}^{(s-1)}\nu_1.$$

Следуя (7.11), запишем для полинома $\mathcal{Q}^{(s+1)}(x)$

$$\mathcal{Q}^{(s+1)}(x) = \mathcal{Q}^{(s)}(x)\,(Ix + \nu_0) + \mathcal{Q}^{(s-1)}(x)\nu_1. \tag{7.16}$$

Применяя данное преобразование t *раз, получим представление для полинома* $\mathcal{Q}^{(s+t)}(x)$ *в виде:*

$$\mathcal{Q}^{(s+t)}(x) = \mathcal{Q}^{(s)}(x)\mathcal{H}_{1s}^{(t)}(x) + \mathcal{Q}^{(s-1)}(x)\mathcal{H}_{0s}^{(t)}(x), \tag{7.17}$$

где $\mathcal{H}_{1s}^{(t)}(x)$ *– матричный полином степени* t, *старший коэффициент которого единичная* $K \times K$ *матрица, а* $\mathcal{H}_{0s}^{(t)}(x)$ *– некоторый матричный полином степени не выше* $t-1$.

Наконец, не сложно проверить, что согласно (7.11) явное вычисление матричных коэффициентов полинома $\mathcal{Q}^{(s+1)}(x)$ *дается следующей*

рекуррентной формулой:

$$\begin{bmatrix} \mathcal{Q}_{s+1}^{(s+1)} \\ \mathcal{Q}_{s}^{(s+1)} \\ \mathcal{Q}_{s-1}^{(s+1)} \\ \mathcal{Q}_{s-2}^{(s+1)} \\ \cdots \\ \mathcal{Q}_{1}^{(s+1)} \\ \mathcal{Q}_{0}^{(s+1)} \end{bmatrix} = \begin{bmatrix} \mathcal{Q}_{s}^{(s)} & 0 & 0 \\ \mathcal{Q}_{s-1}^{(s)} & \mathcal{Q}_{s}^{(s)} & 0 \\ \mathcal{Q}_{s-2}^{(s)} & \mathcal{Q}_{s-1}^{(s)} & \mathcal{Q}_{s-1}^{(s-1)} \\ \mathcal{Q}_{s-3}^{(s)} & \mathcal{Q}_{s-2}^{(s)} & \mathcal{Q}_{s-2}^{(s-1)} \\ \cdots & \cdots & \cdots \\ \mathcal{Q}_{0}^{(s)} & \mathcal{Q}_{1}^{(s)} & \mathcal{Q}_{1}^{(s-1)} \\ 0 & \mathcal{Q}_{0}^{(s)} & \mathcal{Q}_{0}^{(s-1)} \end{bmatrix} \begin{bmatrix} I \\ \nu_0 \\ \nu_1 \end{bmatrix}. \tag{7.18}$$

Заметим, что зная коэффициенты матричного $\mathcal{Q}$-полинома $\mathcal{Q}^{(s+1)}(x)$, можно формально построить блок $Q_{s+1} = \mathcal{Q}^{(s+1)}(A, B)$, который, в силу доказанного ранее, будет A-ортогонален предыдущим $\mathcal{Q}$-блокам Q_0, $\cdots$, Q_s. Таким образом, использование коротких рекуррентных соотношений для вычисления Паде приближений матричного ряда дает альтернативный алгоритм построения A-ортогонального базиса пространства $\mathcal{K}(A, B)$, причем в этом случае блоки $\mathcal{Q}(\mathcal{A}, \mathcal{B})$ A-ортогонального базиса задаются неявно, через коэффициенты соответствующих $\mathcal{Q}$-полиномов.

Подход к вычислению A-ортогонального базиса, основанный на приближениях Паде, имеет видимые преимущества с точки зрения параллельных вычислений. В классическом методе Ланцоша самая «тяжелая» операция алгоритма – это операция умножения матрицы A на блоки Q_i. Однако, как мы увидим, в алгоритме, основанном на вычислении приближений Паде (для краткости будем говорить об алгоритме Ланцоша-Паде), эта операция присутствует лишь при вычислении коэффициентов $\alpha_i = B^T A^i B$ ряда (7.3). Вычисление $K \times K$ матриц α_i не сложно реализовать с использованием K вычислительных узлов, то есть, когда количество узлов совпадает с размером блока. Общий обмен данными при таком вычислении не превосходит cnK (где n – размерность системы, K – размер блока, а c – константа, обозначающая количество машинных слов, необходимое для

хранения одного элемента поля $\mathbb{F}$*), что существенно меньше, чем в случае стандартного метода Ланцоша.*

Проблема однако в том, что мы еще не предъявили способа получения решения X *системы* $AX = B$*. В следующем разделе будет показано как провести оставшиеся вычисления, не вычисляя явным образом* A*-ортогональный базис пространства* $\mathcal{K}(A, B)$*. Новый метод мы будем называть блочным методом Ланцоша-Паде.*

7.4 Блочный метод Ланцоша-Паде

Будем строить алгоритм типа Ланцоша для решения системы уравнений, не вычисляя A*-ортогональных блоков* Q *явно, а работая только с коэффициентами* $\mathcal{Q}$*-полиномов Паде.*

Не сложно проверить, что полиномы $\mathcal{Q}^{(0)}(x)$ *и* $\mathcal{Q}^{(1)}(x)$*, взятые в виде* $\mathcal{Q}^{(0)}(x) = I$ *и* $\mathcal{Q}^{(1)}(x) = Ix - \alpha_1^{-1}\alpha_2$*, с* $\alpha_1 = B^T AB$ *и* $\alpha_2 = B^T A^2 B$*, являются* $\mathcal{Q}$*-полиномами Паде нулевой и первой степени соответственно. Используя частный случай соотношения (7.17), представим* $\mathcal{Q}$*-полином Паде степени* $s+1$ *в виде*

$$\mathcal{Q}^{(s+1)}(x) = \mathcal{Q}^{(1)}(x)\mathcal{H}_1^{(s+1)}(x) + \mathcal{Q}^{(0)}(x)\mathcal{H}_0^{(s+1)}(x), \tag{7.19}$$

с матричным полином $\mathcal{H}_1^{(s+1)}(x)$ *степени* s *и старшим коэффициентом равным единичной матрице и некоторым матричным полином* $\mathcal{H}_0^{(s+1)}(x)$ *степени не выше* $s-1$*. Из (7.16) и (7.19) следует, что* $\mathcal{H}_1^{(s+1)}(x)$ *и* $\mathcal{H}_0^{(s+1)}(x)$ *удовлетворяют одному и тому же рекуррентному соотношению*

$$\mathcal{H}_1^{(s+1)}(x) = \mathcal{H}_1^{(s)}(x)\,(Ix + \nu_0) + \mathcal{H}_1^{(s-1)}(x)\nu_1 \tag{7.20}$$

$$\mathcal{H}_0^{(s+1)}(x) = \mathcal{H}_0^{(s)}(x)\,(Ix + \nu_0) + \mathcal{H}_0^{(s-1)}(x)\nu_1, \tag{7.21}$$

отличаясь только выбором начальных условий. А именно, начальные полиномы выбираются в следующем виде: $\mathcal{H}_1^{(0)}(x) = 0$*,* $\mathcal{H}_1^{(1)}(x) = I$*,* $\mathcal{H}_0^{(0)}(x) = I$*,* $\mathcal{H}_0^{(1)}(x) = 0$*.*

Матричные обозначения *Для удобства дальнейшей записи алгоритмов определим специальные матричные обозначения. Составим из коэффициентов полиномов* $\mathcal{H}_1^{(s)}(x)$ *и* $\mathcal{H}_0^{(s)}(x)$ *блоки* $H_1^{(s)}$ *и* $H_0^{(s)}$ *размера* $n \times K$, *расположив коэффициенты меньших степеней в строках с большими номерами:*

$$H_1^{(s)} = \begin{bmatrix} 0 \\ \cdots \\ 0 \\ \mathcal{H}_{1(s-1)}^{(s)} \\ \mathcal{H}_{1(s-2)}^{(s)} \\ \cdots \\ \mathcal{H}_{11}^{(s)} \\ \mathcal{H}_{10}^{(s)} \end{bmatrix}, \quad H_0^{(s)} = \begin{bmatrix} 0 \\ \cdots \\ 0 \\ 0 \\ \mathcal{H}_{0(s-2)}^{(s)} \\ \cdots \\ \mathcal{H}_{01}^{(s)} \\ \mathcal{H}_{00}^{(s)} \end{bmatrix}. \tag{7.22}$$

В таком случае рекуррентные соотношения (7.20) и (7.21) эквивалентны следующим матричным равенствам:

$$\begin{cases} H_1^{(s+1)} = \begin{bmatrix} ZH_1^{(s)} & H_1^{(s)} & H_1^{(s-1)} \end{bmatrix} \begin{bmatrix} I \\ \nu_0 \\ \nu_1 \end{bmatrix}, \\ H_0^{(s+1)} = \begin{bmatrix} ZH_0^{(s)} & H_0^{(s)} & H_0^{(s-1)} \end{bmatrix} \begin{bmatrix} I \\ \nu_0 \\ \nu_1 \end{bmatrix}, \end{cases} \tag{7.23}$$

где $Z \in \mathbb{F}^{n \times n}$ *– матрица блочного сдвига вида*

$$Z = \begin{bmatrix} 0 & I_k & 0 & \cdots & 0 \\ 0 & 0 & I_k & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & \cdots & I_k \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix}.$$

Наряду с блоками $H_0^{(s)}$ *и* $H_1^{(s)}$ *определим блок* $R^{(s)}$ *размера* $2n \times K$, *который соответствует остаточному ряду Паде степени* s *(заметим, что размер блоков* R *вдвое превосходит размер блоков* H*). Явный вид блока* $R^{(s)}$ *зададим следующим образом:*

$$R^{(s)} = \begin{bmatrix} \star \\ \star \\ \rho^{(s)}_{2\frac{n}{K}-s} \\ \cdots \\ \rho^{(s)}_{s+1} \\ 0 \\ \cdots \\ 0 \end{bmatrix}, \tag{7.24}$$

где символом $\star$ *обозначаются те элементы, которые не являются существенными (другими словами значения этих элементов не будут влиять на вычисления в алгоритме). Число нулевых матриц в* $R^{(s)}$ *в точности равно* s, *и отражает тот факт, что* $R^{(s)}$ *соответствует остаточному ряду Паде степени* s. *По аналогии с формулами (7.23) запишем*

$$R^{(s+1)} = \begin{bmatrix} Z^T R^{(s)} & R^{(s)} & R^{(s-1)} \end{bmatrix} \begin{bmatrix} I \\ \nu_0 \\ \nu_1 \end{bmatrix} \tag{7.25}$$

Заметим, что в соответствии с определением число «несущественных» элементов в $R^{(s+1)}$ *увеличивается на* 1 *по сравнению с числом «несущественных» элементов в* $R^{(s)}$. *Таким образом, формула (7.25) может рассматриваться как равенство только для «существенных» компонент* $R^{(s+1)}$.

Далее, обозначим через Q_1 *и* Q_0 *блоки* $\mathcal{Q}_1(A, B)$ *и* $\mathcal{Q}_0(A, B)$, *соответственно. По определению* $Q_0 = B$, *а* $Q_1 = AB - B\alpha_1^{-1}\alpha_2$. *Составим матрицы* $K_0,\ K_1 \in \mathbb{F}^{n\times n}$ *из блоков вида* $A^j Q_0$ *и* $A^j Q_1$ *с* j *от* 0 *до* $\frac{n}{K} - 1$

записанных в обратном порядке,

$$\begin{aligned} K_0 &= \left[\begin{array}{cccc} A^{\frac{n}{K}-1}Q_0 & A^{\frac{n}{K}-2}Q_0 & \cdots & AQ_0 \quad Q_0 \end{array} \right] \\ K_1 &= \left[\begin{array}{cccc} A^{\frac{n}{K}-1}Q_1 & A^{\frac{n}{K}-2}Q_1 & \cdots & AQ_1 \quad Q_1 \end{array} \right] \end{aligned} \tag{7.26}$$

Из (7.17) следует, что для блока $Q_{s+1} = \mathcal{Q}^{(s+1)}(A,B)$ *справедливо следующее представление:*

$$Q_{s+1} = \sum_{i=0}^{s} A^i Q_1 \mathcal{H}_{1i}^{(s+1)} + \sum_{i=0}^{s-1} A^i Q_0 \mathcal{H}_{0i}^{(s+1)}, \tag{7.27}$$

которое удобно записать в виде

$$Q_{s+1} = K_0 H_0^{(s+1)} + K_1 H_1^{(s+1)}. \tag{7.28}$$

По построению (см. утверждение 1) блоки Q_s *образуют* A*-ортогональный базис всего пространства, а, следовательно, решение системы* $X_{\frac{n}{K}}$ *записывается в виде линейной комбинации блоков* Q_i

$$X_{\frac{n}{K}} = \sum_{i=0}^{\frac{n}{K}-1} Q_i \left(Q_i^T A Q_i\right)^{-1} Q_i^T B.$$

Вводя обозначение Z_i *для матриц*

$$Z_i = \left(Q_i A Q_i^T\right)^{-1} Q_i^T B,$$

придем к краткой записи для $X_{\frac{n}{K}}$

$$X_{\frac{n}{K}} = \sum_{i=0}^{\frac{n}{K}-1} Q_i Z_i. \tag{7.29}$$

Наконец, подставляя в (7.29) выражение для Q_i *из (7.28), найдем, что*

$$\begin{aligned} X_{\frac{n}{K}} &= \sum_{i=0}^{\frac{n}{K}-1} Q_i Z_i = \sum_{i=0}^{\frac{n}{K}-1} \left(K_0 H_0^{(i)} + K_1 H_1^{(i)}\right) Z_i \\ &= K_0 \left(\sum_{i=0}^{\frac{n}{K}-1} H_0^{(i)} Z_i\right) + K_1 \left(\sum_{i=0}^{\frac{n}{K}-1} H_1^{(i)} Z_i\right). \end{aligned}$$

Чтобы еще немного упростить запись и получить матричное соотношение для $X_{\frac{n}{K}}$, *определим систему блоков* $G_0^{(s)}$ *и* $G_1^{(s)}$ *размера* $n \times K$ *следующего вида*

$$G_0^{(s)} = \sum_{i=0}^{s} H_0^{(i)} Z_i = G_0^{(s-1)} + H_0^{(s)} Z_s; \tag{7.30}$$

$$G_1^{(s)} = \sum_{i=0}^{s} H_1^{(i)} Z_i = G_1^{(s-1)} + H_1^{(s)} Z_s, \tag{7.31}$$

где s *меняется в пределах от* 1 *до* $\frac{n}{K} - 1$. *Используя (7.30) и (7.31), окончательно запишем* $X_{\frac{n}{K}}$ *как*

$$X_{\frac{n}{K}} = K_0 G_0^{(\frac{n}{K}-1)} + K_1 G_1^{(\frac{n}{K}-1)}. \tag{7.32}$$

Преимущества полученной записи для $X_{\frac{n}{K}}$ *состоит в том, что в ней нет явной зависимости от блоков* Q_i, *а имеется зависимость лишь от систем* K_0 *и* K_1 *блочно крыловского типа. Вычисление* K_0 *и* K_1 *легко производить на параллельных вычислительных системах. Для этого каждый из* $2K$ *векторов, входящих в блок* Q_0 *или* Q_1 *можно умножать на матрицу* A *независимо, тем самым достигая максимально возможного эффекта. Чтобы завершить описание метода, необходимо предъявить способы вычисления блоков* $G_0^{(\frac{n}{K})}$ *и* $G_1^{(\frac{n}{K})}$, *которые не использовали бы явный вид боков* Q_i.

Вычисление $Q_i^T B$ *Используя для блока* Q_i *выражение*

$$Q_i = K_0 H_0^{(i)} + K_1 H_1^{(i)},$$

получаем, что

$$\begin{aligned} Q_i^T B &= \left(K_0 H_0^{(i)} + K_1 H_1^{(i)}\right)^T B \\ &= (H_0^{(i)})^T K_0^T B + (H_1^{(i)})^T K_1^T B. \end{aligned}$$

Рассмотрим матрицы $K_0^T B$ *и* $K_1^T B$. *Из (7.26) следует, что* $K_0^T B$ *является блоком (то есть матрицей размера* $n \times K$*), составленным из* $K \times K$ *матриц*

$\alpha_i = B^T A^i B$. *А именно,*

$$K_0^T B = \begin{bmatrix} \alpha_{\frac{n}{K}} \\ \cdots \\ \alpha_1 \\ \alpha_0 \end{bmatrix}.$$

Если дополнительно определить матрицы $\beta_i = Q_1^T A^i B$, *то для блока* $K_1^T B$ *будет справедливо аналогичное равенство*

$$K_1^T B = \begin{bmatrix} \beta_{\frac{n}{K}} \\ \cdots \\ \beta_1 \\ \beta_0 \end{bmatrix}.$$

Заметим, что вычисление матриц α_i *и* β_i *можно делать независимо, не изменяя тем самым параллельной сложности алгоритма. Окончательно для* $Q_i^T B$ *получаем следующее выражение:*

$$Q_i^T B = \sum_{j=0}^{i-2} \left(\mathcal{H}_{0j}^{(i)}\right)^T \alpha_j + \sum_{j=0}^{i-1} \left(\mathcal{H}_{1j}^{(i)}\right)^T \beta_j. \tag{7.33}$$

Сложность такого вычисления, как не трудно проверить, оценивается величиной $\mathcal{O}((2i-1)K^3)$. *Следовательно, сложность вычисления всех матриц* $Q_i^T B$ *оценивается как*

$$\left\{ \text{сложность всех } Q_i^T B \right\} = \sum_{i=0}^{\frac{n}{K}} \left(\mathcal{O}((2i-1)K^3)\right) \approx \mathcal{O}(n^2 K).$$

Замечание. *Указанное вычисление можно сделать быстрее по рекуррентной формуле, которая получается из (7.2) транспонированием и умножением справа на* B. *При этом, первое слагаемое в правой части полученного равенства будет равно нулю из-за ортогональности строящихся блоков первому из них.*

Вычисление $Q_i^T A Q_i$

Чтобы вычислить $Q_i^T A Q_i$ *воспользуемся утверждением 1:*

$$Q_i^T A Q_i \ = \left(\mathcal{Q}_i^{(i)}\right)^T \rho_{i+1}^{(i)}$$

Поскольку, старшие коэффициенты $\mathcal{Q}_i^{(i)}$ *у* $\mathcal{Q}$ *полиномов не изменяются с номером* i, *то сложность вычисления матриц* $Q_i^T A Q_i$ *не превосходит*

$$\left\{\text{сложность всех } Q_i^T A Q_i\right\} = \sum_{i=0}^{\frac{n}{K}-1} \left(\mathcal{O}(K^3)\right) = \mathcal{O}(nK^2).$$

Более того, можно считать, что коэффициенты $\mathcal{Q}_i^{(i)}$ *тождественно равны* 1, *и в таком случае вычисление* $Q_i^T A Q_i$ *не требует дополнительных затрат.*

Вычисление Z_i *Поскольку*

$$Z_i = \left(Q_i^T A Q_i\right)^{-1} Q_i^T B,$$

то сложность нахождения всех матриц Z_i *складывается из сложности вычисления всех* $K \times K$ *матриц* $Q_i^T B$, *матриц* $Q_i^T A Q_i$, *а также из сложности вычисления обратных к матрицам* $Q_i^T A Q_i$. *Суммируя все сказанное выше, получаем*

$$\left\{\text{сложность всех } Z_i\right\} = \mathcal{O}\left(n^2 K\right) + \mathcal{O}\left(nK^2\right), \tag{7.34}$$

где второе слагаемое отвечает за операцию обращениия уже вычисленных матриц $Q_i^T A Q_i$ *и операцию произведения матриц* $(Q_i^T A Q_i)^{-1}$ *и* $Q_i^T B$.

Алгоритм Ланцоша-Паде *Выпишем алгоритм Ланцоша-Паде.*

Шаг 0:

(a) вычислить α_0, α_1, α_2;

(b) определить $n \times K$ *блоки* Q_0 *и* Q_1

$$\begin{aligned} Q_0 &= B; \\ Q_1 &= AB - B\alpha_1^{-1}\alpha_2; \end{aligned}$$

(c) определить $n \times K$ *блоки* $H_1^{(0)}$ *и* $H_0^{(0)}$

$$H_1^{(0)} = H_0^{(1)} = \begin{bmatrix} 0 \\ \cdots \\ 0 \\ 0 \end{bmatrix} \qquad H_0^{(0)} = H_1^{(1)} = \begin{bmatrix} 0 \\ \cdots \\ 0 \\ I_k \end{bmatrix}$$

Шаг 1:

(a) вычислить $\alpha_i = Q_0^T A^i Q_0$, $\beta_i = Q_1^T A^i Q_0$ *для всех* i *от* 0 *до* $2\frac{n}{K}$;

(b) присвоить элементам блоков $R_0^{(0)}$ *и* $R_1^{(0)}$ *значения в соответствии со следующими формулами:*

$$R^{(0)} = \begin{bmatrix} \alpha_{2\frac{n}{K}} \\ \alpha_{2\frac{n}{K}-1} \\ \cdots \\ \alpha_2 \\ \alpha_1 \\ 0 \end{bmatrix}, R^{(1)} = \begin{bmatrix} 0 \\ \alpha_{2\frac{n}{K}} - \alpha_{2\frac{n}{K}-1}\alpha_1^{-1}\alpha_2 \\ \cdots \\ \alpha_3 - \alpha_2\alpha_1^{-1}\alpha_2 \\ 0 \\ 0 \end{bmatrix},$$

Шаг 2: *для всех* s *от* 2 *до* $\frac{n}{K}+1$ *повторять (a), (b), (c), (d) и (e)*

(a) найти $K \times K$ *матрицы* ν_0 , ν_1 , *решая систему уравнений (7.14), (7.15);*

(b) вычислить $H_0^{(s)}$ *и* $H_1^{(s)}$ *по формуле (7.23);*

(c) вычислить остаток ряда Паде $R^{(s)}$ *для приближения Паде степени* s *по формуле (7.25);*

(d) вычислить $Z_s = (Q_s^T A Q_s)^{-1} Q_s^T B$

i. вычислить $Q_s^T B$ *по формуле (7.33) или по формуле, полученной из (7.2) транспонированием и умножением справа на* B ;

ii. вычислить $Q_s^T A Q_s$ *по формуле (7.34);*

(e) вычислить $G_0^{(s)}$ *и* $G_1^{(s)}$ *по рекуррентным формулам:*

$$G_0^{(s)} = G_0^{(s-1)} + H_0^{(s)} Z_s, \quad G_1^{(s)} = G_1^{(s-1)} + H_1^{(s)} Z_s. \tag{7.35}$$

Шаг 3: *Вычислить* $X_{\frac{n}{K}}$ *по формуле (7.32).*

Алгоритмическая и параллельная сложность метода Ланцоша-Паде *Вычислительная сложность алгоритма Ланцоша-Паде складывается из сложности нескольких групп вычислений. Первая такая группа вычислений (***Шаг 1 алгоритма Ланцоша-Паде***) связана с вычислениями квадратных* $K \times K$ *матриц* α_i *и* β_i, *где индекс* i *изменяется от* 0 *до* $2\frac{n}{K}$. *Прямой путь к получению этих матриц состоит в том, чтобы сначала последовательно вычислить блоки* Q_0, AQ_0, $\cdots$, $A^{2\frac{n}{K}}Q_0$ *и блоки* Q_1, AQ_1, $\cdots$, $A^{2\frac{n}{K}}Q_1$, *затратив* $\mathcal{O}(4pn^2)$ *операций (*p *– оценка сверху на число ненулей в строках матрицы* A*), а затем определить сами матрицы, затратив еще не более* $\mathcal{O}(n^2K)$ *операций с элементами поля* $\mathbb{F}$. *Несложно понять, что описанные вычисления идеально реализуются на параллельных вычислительных системах.*

Например, рассмотрим систему с K *вычислительными узлами. Будем считать, что перед началом вычислений на каждом узле имеются как матрица системы* A *так и блоки* Q_0 *и* Q_1, *размера* $n \times K$. *Пусть вычислительный узел с номером* J *строит последовательности столбцов* $(Q_0)_J$, $A(Q_0)_J$, $\cdots$, $A^{2\frac{n}{K}}(Q_0)_J$ *и* $(Q_1)_J$, $A(Q_1)_J$, $\cdots$, $A^{2\frac{n}{K}}(Q_1)_J$ *размера* n. *Алгоритмическая сложность такого вычисления не выше* $\mathcal{O}(2p\frac{n^2}{K})$. *Рассматривая выражения вида* $Q_0^T\left(A^i(Q_0)_J\right)$, $Q_0^T\left(A^i(Q_1)_J\right)$, $Q_1^T\left(A^i(Q_1)_J\right)$, *заметим, что в результате очевидных вычислений на* J*-ом узле получаются* J*-ые столбцы матриц* α_i *и* β_i, *причем сложность вычислений на каждом узле не превосходит* $\mathcal{O}(\frac{n^2}{K})$.

Таким образом, с использованием K *вычислительных узлов количество вычислений на одном узле для* **Шага 1** *сократилось в* K *раз.*

Рассмотрим **Шаг 2**. *Сложность решения одной системы линейных уравнений относительно матриц* ν_0, ν_1 *не превосходит* $\mathcal{O}(K^3)$ *операций в поле*

$\mathbb{F}$. *Следовательно, решение всех таких систем требует* $\mathcal{O}(nK^2)$ *операций.*

Далее, для фиксированного s *сложность вычисления блоков* $H_0^{(s)}$ *и* $H_1^{(s)}$ *не более* $\mathcal{O}(nK^2)$. *Следовательно, нахождение всех блоков* H_0 *и* H_1 *потребует не более* $\mathcal{O}(n^2K)$ *действий с элементами поля* $\mathbb{F}$, *и аналогичную сложность имеет вычисление последовательности блоков* $R^{(s)}$.

Наконец, сложность получения матриц Z_s *рассматривалась нами в предыдущем разделе, а сложность получения блоков* $G^{(s)}$, *как нетрудно проверить, оценивается сверху величиной* $\mathcal{O}(n^2K)$. *Отметим, не обсуждая деталей, что как и на* **Шаге 1** *на* **Шаге 2** *все вычисления идеально масштабируются на системах с* K *независимыми узлами, а параллельная сложность* **Шага 2** *в* K *раз меньше алгоритмической сложности.*

В соответствии с формулой (7.32) **Шаг 3** *является ни чем иным как двукратным произведением плотных* $n \times n$ *матриц* K_0 *и* K_1 *на* $n \times K$ *блоки* G_0 *И* G_1. *Таким образом, сложность такого вычисления не выше* $\mathcal{O}(n^2K)$. *Можно показать, что и в этом случае на системе с* K *узлами вычисления можно производить в* K *раз быстрее.*

Собирая вместе сказанное выше, получаем, что сложность блочного метода Ланцоша-Паде с размером блока K *не превосходит*

$$\{\textit{сложность}\} = \mathcal{O}\left(pn^2 + n^2K + nK^2\right), \tag{7.36}$$

причем существует реализация такого метода на вычислительной системе с K *узлами, для которой сложность не выше*

$$\{\textit{параллельная сложность}\} = \mathcal{O}\left(p\frac{n^2}{K} + n^2 + nK\right). \tag{7.37}$$

Заметим, что, как следует из (7.37), параллельная сложность метода Ланцоша-Паде содержит слагаемые, одно из которых n^2 *не зависит от* K, *а другое* nK *растет с размером блока* K. *На практике при малых* K *эти слагаемые малы по сравнению с первым слагаемым. Однако с ростом* K

ситуация меняется. Последнее говорит о том, что предлагаемый подход имеет определенные ограничения при его реализации на параллельных вычислительных системах. Однако основное преимущество рассматриваемой схемы вычислений состоит в сравнительно небольшом обмене данными между узлами (поскольку нет явных вычислений с векторами направлений). По этой причине метод Ланцоша-Паде особенно эффективен для вычислительных систем с медленными межузловыми коммуникациями. Однако, если вычислительный комплекс обладает высокоскоростной коммуникационной сетью, то организацию вычислений в методе Ланцоша-Паде можно изменить, уменьшив параллельную сложность алгоритма за счет определенного увеличения количества обменов. В этом состоит идея универсального метода Ланцоша-Паде, к описанию которого сейчас переходим.

7.5 Универсальный блочный метод Ланцоша-Паде

Относительно высокая алгоритмическая (параллельная) сложность блочного метода Ланцоша-Паде определяется необходимостью проводить вычисления с блоками $H_0^{(s)}$, $H_1^{(s)}$, $R_0^{(s)}$, $R_1^{(s)}$, *размер которых* $n \times K$ *и* $2n \times K$ *Идея универсального метода состоит в переходе к более коротким блокам. Для этого весь процесс вычислений разбивается на несколько последовательных шагов.*

Итак, фиксируем натуральные t *и* q *такие, что* $tq = \frac{n}{K}$. *Для произвольного целого* $p \leq \frac{n}{K}$ *запишем его в виде* $p = jt + s$, *где* s *– остаток от деления* p *на* t, *и, следовательно,* $s < t$. *Принимания во внимание (7.17), полином* $\mathcal{Q}^{(p)}(x)$ *степени* p *будем представлять в виде*

$$\mathcal{Q}^{(p)}(x) = \mathcal{Q}^{(jt+s)}(x) = \mathcal{Q}^{(jt+1)}(x)\mathcal{H}_{1j}^{(s)}(x) + \mathcal{Q}^{(jt)}(x)\mathcal{H}_{0j}^{(t)}(x), \tag{7.38}$$

где, как и следует, степень полинома $\mathcal{H}_{1j}^{(s)}(x)$ *не выше* $s-1$, *а степень полинома* $\mathcal{H}_{0j}^{(s)}$ *не выше* $s-2$. *Будем строить вычисления в алгоритме таким обазом, чтобы использовать полиномы* $\mathcal{H}_{1j}^{(s)}(x)$, $\mathcal{H}_{0j}^{(s)}$. *Очевидно, что вычисления с*

полиномами малых степеней имеют меньшую вычислительную сложность. Это наблюдение и составляет основу универсального метода.

Несложно проверить, что для фиксированного j *полиномы* $\mathcal{H}_{0j}^{(s)}$ *и* $\mathcal{H}_{1j}^{(s)}$ *удовлетворяют одному и тому же рекуррентному соотношению*

$$\mathcal{H}_{1j}^{(s+1)}(x) = \mathcal{H}_{1j}^{(s)}(x)\,(Ix+\nu_0) + \mathcal{H}_{1j}^{(s-1)}(x)\nu_1 \tag{7.39}$$

$$\mathcal{H}_{0j}^{(s+1)}(x) = \mathcal{H}_{0j}^{(s)}(x)\,(Ix+\nu_0) + \mathcal{H}_{0j}^{(s-1)}(x)\nu_1, \tag{7.40}$$

отличаясь только выбором начальных условий. А именно, начальные полиномы выбираются в следующем виде: $\mathcal{H}_{1j}^{(0)}(x) = 0$, $\mathcal{H}_{1j}^{(1)}(x) = I$, $\mathcal{H}_{0j}^{(0)}(x) = I$, $\mathcal{H}_{0j}^{(1)}(x) = 0$. *Конечно, чтобы пользоваться формулами (7.39), (7.40) необходимо еще предоставить способ нахождения для каждого* s *матриц* ν_0 *и* ν_1, *зависящих от* s. *Заметим, что для этих целей мы не можем воспользоваться напрямую формулами (7.24), (7.25). Их использование привело бы к тому, что сложность алгоритма осталась бы неизменной. Для того, чтобы избавиться от необходимости работать с большими блоками* $R^{(p)}$ *нам потребуется следующее вспомогательное утверждение.*

Лемма 3 *Пусть* $l > t > 0$, *и пусть среди всех коэффициентов* $\mathcal{Q}$*-полинома Паде* $\mathcal{Q}^{(l)}(x)$ *известны только первые* t, *тогда коэффициенты* $\rho_{l+1}^{(l)}$, $\rho_{l+2}^{(l)}$, $\cdots$, $\rho_{l+t}^{(l)}$ *остаточного ряда Паде* $\mathcal{R}^{(l)}(x)$ *могут быть определены как решения системы линейных с нижней треугольной блочной теплицевой матрицей*

$$\begin{bmatrix} \mathcal{Q}_l^{(l)T} & & & & \\ \mathcal{Q}_{l-1}^{(l)T} & \mathcal{Q}_l^{(l)T} & & & \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \mathcal{Q}_{l-t+2}^{(l)T} & \mathcal{Q}_{l-t+3}^{(l)T} & & \mathcal{Q}_l^{(l)T} & \\ \mathcal{Q}_{l-t+1}^{(l)T} & \mathcal{Q}_{l-t+2}^{(l)T} & & \mathcal{Q}_{l-1}^{(l)T} & \mathcal{Q}_l^{(l)T} \end{bmatrix} \begin{bmatrix} \rho_{l+1}^{(l)} \\ \rho_{l+2}^{(l)} \\ \cdots \\ \rho_{l+t-1}^{(l)} \\ \rho_{l+t}^{(l)} \end{bmatrix} = \begin{bmatrix} Q_l^T A Q_l \\ Q_l^T A^2 Q_l \\ \cdots \\ Q_l^T A^{t-1} Q_l \\ Q_l^T A^t Q_l \end{bmatrix}, \tag{7.41}$$

где Q_l *– блок, соответствующий полиному* $\mathcal{Q}^{(l)}(x)$, $Q_l = \mathcal{Q}^{(l)}(A,B)$.

Доказательство. *Пусть* $\mathcal{Q}^{(l)}(x)$ *–* $\mathcal{Q}$*-полином Паде степени* l

$$\mathcal{Q}^{(l)}(x) = \mathcal{Q}_l^{(l)}x^l + \mathcal{Q}_{l-1}^{(l)}x^{l-1} + \cdots \mathcal{Q}_1^{(l)}x + \mathcal{Q}_0^{(l)}.$$

По определению (см. раздел 7.2) значение билинейного отображения $< \mathcal{Q}^{(l)}(x), \mathcal{Q}^{(l)}(x) >$ *является коэффициентом при* x^{-1} *формального ряда* $\left(\mathcal{Q}^{(l)}(x)\right)^T \alpha(x)\mathcal{Q}^{(l)}(x)$. *Так как* $\mathcal{Q}^{(l)}$ *– это* $\mathcal{Q}$*-полином Паде, то*

$$\begin{aligned} < \mathcal{Q}^{(l)}(x), \mathcal{Q}^{(l)}(x) > &= coeff_{-1}\left(\left(\mathcal{Q}^{(l)}(x)\right)^T \alpha(x)\mathcal{Q}^{(l)}(x)\right) \\ &= coeff_{-1}\left(\left(\mathcal{Q}^{(l)}(x)\right)^T \left(\alpha(x)\mathcal{Q}^{(l)}(x) - \mathcal{P}^{(l)}(x)\right)\right) \\ &= coeff_{-1}\left(\left(\mathcal{Q}^{(l)}(x)\right)^T \left(\sum_{i=l+1}^{\infty} \rho_i^{(l)} x^{-i}\right)\right) \\ &= (\mathcal{Q}_l^{(l)})^T \rho_{l+1}^{(l)}. \end{aligned}$$

С другой стороны, в соответствии с леммой 1

$$< \mathcal{Q}^{(l)}(x), \mathcal{Q}^{(l)}(x) >= Q_l^T A Q_l.$$

Сопоставляя последние равенства, получаем первую строку в уравнении (7.41). Чтобы получить k*-ую строку в уравнении (7.41), рассмотрим билинейное отображение на полиномах* $x^k\mathcal{Q}^{(l)}(x)$ *и* $\mathcal{Q}^{(l)}(x)$. *Используя теже преобразования, получим*

$$\begin{aligned} < x^k\mathcal{Q}^{(l)}(x), \mathcal{Q}^{(l)}(x) > &= coeff_{-1}\left(\left(x^k\mathcal{Q}^{(l)}(x)\right)^T \alpha(x)\mathcal{Q}^{(l)}(x)\right) \\ &= coeff_{-1}\left(\left(x^k\mathcal{Q}^{(l)}(x)\right)^T \left(\sum_{i=l+1}^{\infty} \rho_i^{(l)} x^{-i}\right)\right) \\ &= (\mathcal{Q}_{l-k}^{(l)})^T \rho_{l+1}^{(l)} + (\mathcal{Q}_l^{(l-k+1)})^T \rho_{l+2}^{(l)} + \cdots + (\mathcal{Q}_l^{(l)})^T \rho_{l+k}^{(l)}. \end{aligned}$$

Остается только заметить, что в соответствии с леммой 1 из раздела 7.2

$$< x^k\mathcal{Q}^{(l)}(x), \mathcal{Q}^{(l)}(x) >= Q_l^T A^{k+1} Q_l.$$

□

С помощью леммы 3 вычисления полиномов $\mathcal{H}_{0j}^{(s)}$ *и* $\mathcal{H}_{1j}^{(s)}$ *будем производить по следующей схеме.*

Будем предполагать, что для некоторого фиксированного j одновременно известны блоки $Q_{jt} = \mathcal{Q}^{(jt)}(A,B)$ и $Q_{jt+1} = \mathcal{Q}^{(jt+1)}(A,B)$, а также $2(t+1)$ старших коэффициентов матричных полиномов $\mathcal{Q}^{(jt)}(x)$ и $\mathcal{Q}^{(jt+1)}(x)$. Поскольку блок Q_{jt} известен явно то можно найти матрицы $\alpha_{jt,s} = Q_{jt}^T A^s Q_{jt}$, где s изменяется от 0 до $2(t+1)$. Решая систему линейных уравнений (7.41), найдем $2(t+1)$ старших коэффициентов $\rho_{jt+s}^{(jt)}$ остатка ряда Паде приближения Паде степени jt. Используя найденную часть остатка ряда Паде, вычисляем матричные полиномы $\mathcal{H}_{0j}^{(s)}(x)$ и $\mathcal{H}_{1j}^{(s)}(x)$ для всех s от 0 до $t+1$.

Матричные обозначения *Фиксируем натуральные t и q такие, что $tq = \frac{n}{K}$. Определим квадратные $n \times n$ матрицы K_0 и K_1 следующего вида:*

$$K_0 = \left[\begin{array}{cccc} K_{q-1}^0 & K_{q-2}^0 & \cdots & K_0^0 \end{array} \right], \quad K_1 = \left[\begin{array}{cccc} K_{q-1}^1 & K_{q-2}^1 & \cdots & K_0^1 \end{array} \right], \tag{7.42}$$

где $n \times tK$ матрицы K_j^0 и K_j^1, $j = 0, \cdots, q-1$ являются цепочками из крыловских блоков:

$$K_j^0 = \left[\begin{array}{cccc} A^{t-1}Q_{tj} & A^{t-2}Q_{tj} & \cdots & AQ_{tj} \quad Q_{tj} \end{array} \right] \tag{7.43}$$

$$K_j^1 = \left[\begin{array}{cccc} A^{t-1}Q_{tj+1} & A^{t-2}Q_{tj+1} & \cdots & AQ_{tj+1} \quad Q_{tj+1} \end{array} \right] \tag{7.44}$$

Несложно понять, что формулы (7.42), (7.43) и (7.44) задают базис пространства Крылова $\mathcal{K}(A,B)$. Блоки K_j^0 и K_j^1 сами являются цепочками из блоков крыловского типа. Можно сказать, что K_0 и K_1 – это кусочно крыловские базисы пространства $\mathcal{K}(A,B)$.

Помимо матриц K_j^0 и K_j^1, определим $n \times (t+1)$ матрицы $\hat{K}_j^0$ и $\hat{K}_j^1$ вида

$$\hat{K}_j^0 = \left[\begin{array}{cccc} A^tQ_{tj} & A^{t-1}Q_{tj} & \cdots & AQ_{tj} \quad Q_{tj} \end{array} \right] \tag{7.45}$$

$$\hat{K}_j^1 = \left[\begin{array}{cccc} A^tQ_{tj+1} & A^{t-1}Q_{tj+1} & \cdots & AQ_{tj+1} \quad Q_{tj+1} \end{array} \right]. \tag{7.46}$$

Представим произвольное $p \leq \frac{n}{K}$ в виде $p = jt+s$, где s – остаток от деленния p на t, $s < t$. Тогда, принимания во внимание (7.17),

$$\mathcal{Q}^{(p)}(x) = \mathcal{Q}^{(jt+s)}(x) = \mathcal{Q}^{(jt+1)}(x)\mathcal{H}_{1j}^{(s)}(x) + \mathcal{Q}^{(jt)}(x)\mathcal{H}_{0j}^{(s)}(x), \tag{7.47}$$

причем степень полиномов $\mathcal{H}_{1j}^{(s)}(x)$ *не выше* $s-1$, *а степень полинома* $\mathcal{H}_{0j}^{(s)}$ *не выше* $s-2$. *Определим* $(t+1)K \times K$ *блоки* $H_{0j}^{(s)}$ *и* $H_{1j}^{(s)}$, *элементами которых будут* $K \times K$ *матрицы коэффициентов полиномов* $\mathcal{H}_{0j}^{(s)}$ *и* $\mathcal{H}_{1j}^{(s)}$,

$$H_{1j}^{(s)} = \begin{bmatrix} 0 \\ \cdots \\ 0 \\ \mathcal{H}_{1j(s-1)}^{(s)} \\ \mathcal{H}_{1j(s-2)}^{(s)} \\ \cdots \\ \mathcal{H}_{1j1}^{(s)} \\ \mathcal{H}_{1j0}^{(s)} \end{bmatrix}, \quad H_{0j}^{(s)} = \begin{bmatrix} 0 \\ \cdots \\ 0 \\ 0 \\ \mathcal{H}_{0j(s-2)}^{(s)} \\ \cdots \\ \mathcal{H}_{0j1}^{(s)} \\ \mathcal{H}_{00}^{(s)} \end{bmatrix}. \tag{7.48}$$

В таком случае для блока $Q_p = \mathcal{Q}^{(p)}(A,B)$ *справедливо следующее представление*

$$Q_p = Q_{jt+s} = \hat{K}_j^0 H_{0j}^{(s)} + \hat{K}_j^1 H_{1j}^{(s)}. \tag{7.49}$$

Существенным отличием (7.49) от (7.28) является размер входящих в них матриц.

Замечание 1. *Размер блоков* $H_{0j}^{(s)}$ *и* $H_{1j}^{(s)}$ *был выбран таким образом, чтобы иметь возможность представления блоков* $Q_{t(j+1)}$ *и* $Q_{t(j+1)+1}$ *через блоки* Q_{tj} *и* Q_{tj+1}

$$\begin{aligned} Q_{t(j+1)} &= \hat{K}_j^0 H_{0j}^{(t)} + \hat{K}_j^1 H_{1j}^{(t)} \\ Q_{t(j+1)+1} &= \hat{K}_j^0 H_{0j}^{(t+1)} + \hat{K}_j^1 H_{1j}^{(t+1)} \end{aligned}$$

Вслед за блоками $H_{0j}^{(s)}$, $H_{1j}^{(s)}$ *определим блок* $R_j^{(s)}$ *размера* $2(t+1)K \times K$

$$R_j^{(s)} = \begin{bmatrix} \star \\ \star \\ \rho_{2(t+1)-s}^{(jt+s)} \\ \cdots \\ \rho_{s+1}^{(jt+s)} \\ 0 \\ \cdots \\ 0 \end{bmatrix}, \tag{7.50}$$

где символом $\star$ *обозначаются те элементы, которые не являются существенными (другими словами значения этих элементов не будут влиять на вычисления в алгоритме). Число нулевых матриц в блоке* $R_j^{(s)}$ *равно* s. *Если для некоторого* s *известны матрицы* ν_0 *и* ν_1, *то следствием формул (7.39) и (7.40) будут следующие соотношения для блоков* $H_{0j}^{(s+1)}$ *и* $H_{1j}^{(s+1)}$ *:*

$$\begin{cases} H_{1j}^{(s+1)} = \begin{bmatrix} ZH_{1j}^{(s)} & H_{1j}^{(s)} & H_{1j}^{(s-1)} \end{bmatrix} \begin{bmatrix} I \\ \nu_0 \\ \nu_1 \end{bmatrix}, \\ H_{0j}^{(s+1)} = \begin{bmatrix} ZH_{0j}^{(s)} & H_{0j}^{(s)} & H_{0j}^{(s-1)} \end{bmatrix} \begin{bmatrix} I \\ \nu_0 \\ \nu_1 \end{bmatrix}, \end{cases} \tag{7.51}$$

с матрицей блочного сдвига $Z \in \mathbb{F}^{K(t+1)\times K(t+1)}$ *вида*

$$Z = \begin{bmatrix} 0 & I_k & 0 & \cdots & 0 \\ 0 & 0 & I_k & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & \cdots & I_k \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix}.$$

В тоже время блок $R_j^{(s+1)}$ *определяется соотношением*

$$R^{(s+1)} = \left[\begin{array}{ccc} Z^T R^{(s)} & R^{(s)} & R^{(s-1)} \end{array} \right] \left[\begin{array}{c} I \\ \nu_0 \\ \nu_1 \end{array} \right] \tag{7.52}$$

Заметим, что в соответствии с определением число «несущественных» элементов в $R_j^{(s+1)}$ *увеличивается на* 1 *по сравнению с числом «несущественных» элементов в* $R_j^{(s)}$. *Таким образом, формула (7.52) может рассматриваться как равенство только для «существенных» компонент* $R_j^{(s+1)}$.

Так как для решения системы $X_{\frac{n}{K}}$ *справедливо представление*

$$X_{\frac{n}{K}} = \sum_{i=0}^{\frac{n}{K}-1} Q_i Z_i, \tag{7.53}$$

с матрицами $Z_i = \left(Q_i A Q_i^T\right)^{-1} Q_i^T B$, *то подставив в (7.53) выражение для* Q_i *из (7.49), найдем, что*

$$\begin{aligned} X_{\frac{n}{K}} &= \sum_{i=0}^{\frac{n}{K}-1} Q_i Z_i = \sum_{j=0}^{\frac{n}{Kt}-1} \sum_{s=0}^{t-1} \left(\hat{K}_j^0 H_{0j}^{(s)} + \hat{K}_j^1 H_{1j}^{(s)} \right) Z_{jt+s} \\ &= \sum_{j=0}^{\frac{n}{Kt}-1} \hat{K}_j^0 \left(\sum_{s=0}^{t-1} H_{0j}^{(s)} Z_{jt+s} \right) + \sum_{j=0}^{\frac{n}{Kt}-1} \hat{K}_j^1 \left(\sum_{s=0}^{t-1} H_{1j}^{(s)} Z_{jt+s} \right) \\ &= \sum_{j=0}^{\frac{n}{Kt}-1} \left(\hat{K}_j^0 G_{0j}^{(t-1)} + \hat{K}_j^1 G_{1j}^{(t-1)} \right) \\ &= \sum_{j=0}^{\frac{n}{Kt}-1} \hat{X}_j, \end{aligned}$$

где $(t+1)K \times K$ *блоки* $G_{0j}^{(s)}$ *и* $G_{1j}^{(s)}$ *определяются с помощью следующих*

соотношений

$$G_{0j}^{(s)} = \sum_{i=0}^{s} H_{0j}^{(i)} Z_{jt+s} = G_{0j}^{(s-1)} + H_{0j}^{(s)} Z_{jt+s}, \tag{7.54}$$

$$G_{1j}^{(s)} = \sum_{i=0}^{s} H_{1j}^{(i)} Z_{jt+s} = G_{1j}^{(s-1)} + H_{1j}^{(s)} Z_{jt+s}, \tag{7.55}$$

с параметром s, *меняющимся в пределах от* 1 *до* $t-1$, *а* $n \times K$ *блоки* $\hat{X}_j$ *ни что иное как*

$$\hat{X}_j = \hat{K}_j^0 G_{0j}^{(t-1)} + \hat{K}_j^1 G_{1j}^{(t-1)}.$$

Чтобы завершить описание универсального метода Ланцоша-Паде и оценить его сложность (параллельную сложность), требуется предъявить способы вычисления основных величин алгоритма, как это было сделано в случае метода Ланцоша-Паде из раздела 7.4.

Вычисление $Q_i^T B$ *Пусть* $i = jt + s$, *где* $0 \le s < t$ *– остаток. Используя матричное представление для* Q_i,

$$Q_i = \hat{K}_j^0 H_{0j}^{(s)} + \hat{K}_j^1 H_{1j}^{(s)}, \tag{7.56}$$

запишем

$$Q_i^T B = Q_{jt+s} = (H_{0j}^{(s)})^T \hat{K}_j^{0T} B + (H_{1j}^{(s)})^T \hat{K}_j^{1T} B.$$

Анализ равенства начнем с блоков $\hat{K}_j^{0T} B$ *и* $\hat{K}_j^{1T} B$. *Размер каждого из блоков* $\hat{K}_j^{0T} B$ *и* $\hat{K}_j^{1T} B$ *равен* $(t+1)K \times K$, *а сами блоки имеют весьма специальный вид:*

$$\hat{K}_j^{0T} B = \begin{bmatrix} Q_{jt}^T A^t B \\ Q_{jt}^T A^{t-1} B \\ \cdots \\ Q_{jt}^T B \end{bmatrix}, \quad \hat{K}_j^{1T} B = \begin{bmatrix} Q_{jt+1}^T A^t B \\ Q_{jt+1}^T A^{t-1} B \\ \cdots \\ Q_{jt+1}^T B \end{bmatrix}. \tag{7.57}$$

Будем различать два случая, когда $j = 0$ *и когда* $j \neq 0$. *Если* $j = 0$, *то* $Q_{jt} = Q_0$, *а* $Q_{jt+1} = Q_1$. *Чтобы найти* $\hat{K}_0^{0T} B$ *и* $\hat{K}_0^{1T} B$ *надо вычислять значения*

билинейных форм α_i *и* β_i *для всех* i *от* 0 *до* $t-1$. *Окончательно,*

$$Q_i^T B = \sum_{j=0}^{i-2} \left(\mathcal{H}_{0j}^{(i)}\right)^T \alpha_j + \sum_{j=0}^{i-1} \left(\mathcal{H}_{1j}^{(i)}\right)^T \beta_j, \tag{7.58}$$

для всех $i < t$.

Если $j > 0$, *то вычисления упрощаются. Покажем, что в блоках* $\hat{K}_j^{0T} B$ *и* $\hat{K}_j^{1T} B$ *только матрицы* $Q_{jt}B$ *и* $Q_{jt+1}^T B$ *являются ненулевыми. Действительно, в силу построения, блок* $Q_{jt} = \mathcal{Q}^{(jt)}(A,B)$ *является* A*-отогональным для всех блоков из линейной оболочки* $< B, AB, \cdots, A^{jt-1}B >$, *а значит и для всех блоков* $< B, AB, \cdots, A^{t-1}B >$. *Аналогично, блок* Q_{jt+1} *является* A*-ортогональным ко всем блокам* $< B, AB, \cdots, A^{jt}B >$, *а значит и для любого блока* $< B, AB, \cdots, A^{t-1}B >$. *Учитывая сказанное, представим блоки (7.57) в виде*

$$\hat{K}_j^{0T} B = \begin{bmatrix} 0 \\ 0 \\ \cdots \\ Q_{jt}^T B \end{bmatrix}, \quad \hat{K}_j^{1T} B = \begin{bmatrix} 0 \\ 0 \\ \cdots \\ Q_{jt+1}^T B \end{bmatrix}. \tag{7.59}$$

Таким образом, для выражения $Q_i^T B = Q_{jt+s}^T B$, *в случае* $t > 0$ *верно следующее равенство:*

$$Q_i^T B = Q_{jt+s}^T B = \mathcal{H}_{0j0}^{(s)T} Q_{jt}^T B + \mathcal{H}_{1j0}^{(s)T} Q_{jt+1}^T B.$$

Сложность такого вычисления ограничена сверху величиной $\mathcal{O}(K^3)$. *Следовательно, сложность вычисления всех таких выражений для* $i = jt + s$ *с* $t > 0$ *определяется как* $\mathcal{O}\left(nK^2\right)$. *Для* $j = 0$ *сложность вычислений всех выражений* $Q_i^T B$ *ограничивается величиной* $\mathcal{O}(t^2K^3) = \mathcal{O}(\frac{n^2K}{q^2})$. *Окончательно,*

$$\left\{ \text{ сложность вычисления } Q_i^T B \right\} = \mathcal{O}\left(nK^2 + \frac{n^2K}{q^2}\right). \tag{7.60}$$

Замечание. *Также как и в разделе 5, этом месте проще воспользоваться рекуррентной формулой (7.2).*

Вычисление $Q_i^T A Q_i$ *Вычисление* $Q_i^T A Q_i$ *в случае универсального алгоритма ничем не отличается от вычисления, представленного ранее в предыдущем разделе. Следовательно, сохраняется и сложность такого вычисления*

$$\left\{ \textit{сложность всех } Q_i^T A Q_i \right\} = \mathcal{O}\left(nK^2\right).$$

Вычисление Z_i *Поскольку*

$$Z_i = \left(Q_i^T A Q_i\right)^{-1} Q_i^T B,$$

то сложность нахождения всех матриц Z_i *складывается из сложности вычисления всех* $K \times K$ *матриц* $Q_i^T B$, *матриц* $Q_i^T A Q_i$, *а также из сложности вычисления обратных к матрицам* $Q_i^T A Q_i$. *Таким образом,*

$$\left\{ \textit{сложность всех } Z_i \right\} = \mathcal{O}\left(\frac{n^2 K}{q^2} + nK^2\right), \tag{7.61}$$

где второе слагаемое отвечает за операцию обращениия уже вычисленных матриц $Q_i^T A Q_i$ *и операцию произведения матриц* $(Q_i^T A Q_i)^{-1}$ *и* $Q_i^T B$.

Описание универсального алгоритма Ланцоша-Паде *Фиксируем положительные целые* t *и* K. *Будем считать, что* n *делится на произведение* tK *без остатка и найдется такое целое* q, *что* $q = \frac{n}{tK}$. *Выпишем универсальный алгоритм Ланцоша-Паде.*

Шаг 0:

(a) положить $j = 0$

(b) вычислить α_0, α_1, α_2;

(c) положить

$$\begin{aligned} Q_0 &= B; \\ Q_1 &= AB - B\alpha_1^{-1}\alpha_2; \end{aligned}$$

(d) положить $(t+1)K$ *блоки* $H_{10}^{(0)}$, $H_{00}^{(0)}$, $H_{10}^{(1)}$, $H_{00}^{(1)}$ *равными*

$$H_{10}^{(0)} = H_{00}^{(1)} = \begin{bmatrix} 0 \\ \cdots \\ 0 \\ 0 \end{bmatrix}, \quad H_{00}^{(0)} = H_{10}^{(1)} = \begin{bmatrix} 0 \\ \cdots \\ 0 \\ I_k \end{bmatrix};$$

(e) определить блоки $Q^{0,0}$, $Q^{1,0}$ *размера* $2(t+1)\times K$, *компоненты которых старшие* $2t+1$ *коэффициентов полиномов* $\mathcal{Q}^{(0)}(x)$ *и* $\mathcal{Q}^{(1)}(x)$ *в виде*

$$Q^{1,0} = \begin{bmatrix} 0 \\ \cdots \\ I_k \\ -\alpha_1^{-1}\alpha_2 \end{bmatrix}, \quad Q^{0,0} = \begin{bmatrix} 0 \\ \cdots \\ 0 \\ I_k \end{bmatrix}.$$

Шаг 1:

(a) вычислить $\hat{K}_0^0$ *и* $\hat{K}_0^1$

(b) вычислить $\alpha_i = Q_0^T A^i Q_0$, $\beta_i = Q_1^T A^i Q_0$, *для всех* i *от* 0 *до* $2(t+1)$;

(c) присвоить элементам блоков $R_0^{(0)}$ *и* $R_0^{(1)}$ *значения в соответствии со следующими формулами:*

$$R_0^{(0)} = \begin{bmatrix} \alpha_{0,2(t+1)} \\ \alpha_{0,2t+1} \\ \cdots \\ \alpha_{0,2} \\ \alpha_{0,1} \\ 0 \end{bmatrix}, R_0^{(1)} = \begin{bmatrix} 0 \\ \alpha_{0,2(t+1)} - \alpha_{0,2t+1}\alpha_{0,1}^{-1}\alpha_{0,2} \\ \cdots \\ \alpha_{0,3} - \alpha_{0,2}\alpha_{0,1}^{-1}\alpha_{0,2} \\ 0 \\ 0 \end{bmatrix}, \qquad (7.62)$$

Шаг 2: *Loop* $s := 2, \cdots, t+1$

(a) найти $K\times K$ *матрицы* ν_0, ν_1, *решая систему уравнений (7.14), (7.15);*

(b) вычислить $H_{00}^{(s)}$ *и* $H_{10}^{(s)}$;

(c) вычислить остаток ряда Паде $R_0^{(s)}$ *для приближения Паде степени* s *по формуле (7.52);*

(d) вычислить блок $Q^{s,0}$*, состоящий из* $2(t+1)$ *старших коэффициентов полинома* $\mathcal{Q}^{(s)}$*;*

(e) вычислить $K \times K$ *матрицы* Z_s*, используя соотношение* $Z_s = (Q_s^T A Q_s)^{-1} Q_s^T B$*:*

i. вычислить $Q_s^T B$ *по формуле (7.58);*

ii. вычислить $Q_s^T A Q_s$ *по формуле (7.34);*

(f) вычислить $G_0^{(s)}$ *и* $G_1^{(s)}$*:*

$$G_{00}^{(s)} = G_{00}^{(s-1)} + H_{00}^{(s)} Z_s, \quad G_{10}^{(s)} = G_{10}^{(s-1)} + H_{10}^{(s)} Z_s. \tag{7.63}$$

Шаг 3: *Вычислить* $\hat{X}_0$ *по формуле*

$$\hat{X}_0 = \hat{K}_0^0 G_{00}^{(t-1)} + \hat{K}_0^1 G_{10}^{(t-1)}$$

Шаг 4: *Loop* $j := 1, \cdots, \frac{n}{Kt} - 1$

(a) вычислить $n \times K$ *блоки* $Q_{j(t+1)}$ *и* $Q_{j(t+1)+1}$ *по формуле*

$$\begin{aligned} Q_{j(t+1)} &= \hat{K}_j^0 H_{0j}^{(t)} + \hat{K}_j^1 H_{1j}^{(t)}, \\ Q_{j(t+1)+1} &= \hat{K}_j^0 H_{0j}^{(t+1)} + \hat{K}_j^1 H_{1j}^{(t+1)}. \end{aligned}$$

(b) вычислить $\hat{K}_j^0$ *и* $\hat{K}_j^1$

(c) вычислить $\alpha_i = Q_{jt}^T A^i Q_{jt}$*,* $\beta_i = Q_{jt+1}^T A^i Q_{jt}$*, для всех* i *от* 0 *до* $2(t+1)$*;*

(d) вычислить $2(t+1) \times K$ *блоки* $R_j^{(0)}$ *и* $R_j^{(1)}$*, используя лемму 3 из раздела 7.5*

(e) Loop $s := 2, \cdots, t+1$

i. найти $K \times K$ *матрицы* ν_0*,* ν_1*, решая систему уравнений (7.14), (7.15);*

ii. вычислить $H_{0j}^{(s)}$ *и* $H_{1j}^{(s)}$ *по формуле (7.51);*

iii. вычислить остаток ряда Паде $R_j^{(s)}$ *для приближения Паде степени* s *по формуле (7.52);*

iv. вычислить блок $Q^{s,j}$, *состоящий из* $2(t+1)$ *старших коэффициентов полинома* $\mathcal{Q}^{(jt+s)}$;

v. вычислить $K \times K$ *матрицы* Z_{jt+s}, *используя соотношение* $Z_{jt+s} = (Q_{jt+s}^T A Q_{jt+s})^{-1} Q_{jt+s}^T B$:

A. вычислить $Q_{jt+s}^T B$;

B. вычислить $Q_{jt+s}^T A Q_{jt+s}$ *по формуле (7.34);*

vi. вычислить $G_{0j}^{(s)}$ *и* $G_{1j}^{(s)}$:

$$G_{0j}^{(s)} = G_{0j}^{(s-1)} + H_{0j}^{(s)} Z_{jt+s}, \quad G_{1j}^{(s)} = G_{1j}^{(s-1)} + H_{1j}^{(s)} Z_{jt+s}. \tag{7.64}$$

(f) Вычислить $\hat{X}_j$ *по формуле*

$$\hat{X}_j = \hat{K}_j^0 G_{0j}^{(t-1)} + \hat{K}_j^1 G_{1j}^{(t-1)}$$

Шаг 5: *Вычислить* $X_{\frac{n}{K}}$ *по формуле*

$$X_{\frac{n}{K}} = \hat{X}_0 + \hat{X}_1 + \cdots + \hat{X}_{q-1}.$$

Сложность универсального метода Ланцоша-Паде

Анализ сложности универсального метода Ланцоша-Паде мало чем отличается от анализа сложности простого метода Ланцоша-Паде. Большая часть вычислений в универсальном методе Ланцоша-Паде является калькой с вычислений в простом методе Ланцоша-Паде. Тем не менее, существуют определенные отличия.

Во-первых, вычисления с блоками $H_{0j}^{(s)}$, $H_{1j}^{(s)}$ *и* $R_{0j}^{(s)}$, $R_{1j}^{(s)}$ *становятся проще, так как блоки становятся короче. Действительно, для фиксированного* j *сложность вычислений с такими блоками для* s *от* 0 *до* $t+1$ *не превосходит*

$\mathcal{O}(t^2K^3)$, *а поскольку сам параметр* j *изменяется в пределах от* 0 *до* $\frac{n}{Kt}$, *то сложность общей работы с блоками не превосходит* $\mathcal{O}(ntK^2) = \mathcal{O}(\frac{n^2K}{q})$, *что в* q *раз меньше, чем в случае простого метода Ланцоша-Паде.*

Во-вторых, по сравнению с методом Ланцоша-Паде в универсальный алгоритм добавляется специализированная процедура вычисления коэффициентов остатка ряда Паде длины $2(t+1)$ *для полиномов вида* $\mathcal{Q}^{(jt)}(x)$ *с* j *от* 0 *до* $q-1$. *Сложность этой процедуры определяется сложностью решения системы линейных уравнений с нижней треугольной блочно теплицевой матрицей размера* $2(t+1) \times 2(t+1)$ *(см. лемму 3). Сложность одного такого вычисления не выше* $\mathcal{O}(t^2K^3)$. *Всего же таких вычислений* q, $q = \frac{n}{Kt}$. *Следовательно, общая сложность всех таких вычислений* $\mathcal{O}(ntK^2) = \mathcal{O}(\frac{n^2K}{q})$.

Кроме того, в универсальном алгоритме требуется вычислять $2(t+1)$ *старших коэффициентов всех* $\mathcal{Q}$*-полиномов. Сложность этой группы вычислений, как нетрудно проверить, также оценивается выражением* $\mathcal{O}(\frac{n^2K}{q})$.

Наконец, остальные вычисления и их сложность рассматривались нами в этом разделе выше. Собирая вместе оценки сложности, приходим к утверждению, что для сложности универсального метода справедливо соотношение

$$\{\text{ сложность универсального метода}\} = \mathcal{O}\left(pn^2 + \frac{n^2K}{q} + nK^2\right). \qquad (7.65)$$

Не вдаваясь в детали, укажем, что не сложно предъявить способ параллельных вычислений на системе с K *узлами, для которой параллельная сложность универсального алгоритма Ланцоша-Паде примет вид*

$$\{\text{ параллельная сложность метода}\} = \mathcal{O}\left(\frac{pn^2}{K} + \frac{n^2}{q} + nK\right). \qquad (7.66)$$

Сравнивая (7.66) с (7.37), отметим, что в универсальном методе слагаемое

вида n^2 *переходит в слагаемое вида* $\frac{n^2}{q}$, *что и делает новый метод более эффективным.*

Чем же мы платим? Напомним, что универсальный метод предполагает явное вычисление блоков вида Q_{jt} *и* Q_{jt+1}. *Каждое такое вычисление связано с большим обменом данными между вычислительными узлами. Чем меньше* t, *тем чаще происходит такой обмен, и время на обмен увеличивается. С другой стороны, чем меньше* t, *тем больше* q *и тем меньше параллельная сложность алгоритма.*

Часть IV

Заключение

В диссертации разработан ряд новых алгоритмов существенно упрощающих задачи дискретного логарифмирования и факторизации и некоторых смежных задач, а также получены новые характеристики существующих алгоритмов.

В первой главе доказана полиномиальная эквивалентность сертифицированной задачи дискретного логарифмирования и задачи Диффи-Хеллмэна при применении алгоритмов, использующих операции не сложнее групповой и работающих в исходной группе и связанных с нею по дереву Пратта группах, в случае, когда растущий порядок исходной группы имеет дерево Пратта с ограниченными длинами ветвей. В общем случае, получена формула, связывающая сложности этих задач. Предложенный в данной главе подход применён к анализу стойкости Российского стандарта цифровой подписи ГОСТ Р 34.10-2012 эллиптических кривых. Показано, что в случае, когда $p-1$ *"хорошо"раскладывается на простые множители, задачи дискретного логарифмирования и Диффи-Хеллмэна в группе простого порядка* p*, на стандартных эллиптических кривых, полиномиально эквивалентны. Предложено несколько подходов и алгоритмов решения задачи дискретного логарифмирования на стандартных эллиптических кривых. Данные результы могут свидетельствать о слабости действующего стандарта эллиптических кривых. Предложены способы его усовершенствования.*

Во второй главе получены оценки на величину и количество простых делителей чисел вида $p-1$*, которые характеризуют дерево Пратта и могут быть применены к описанным атакам на ГОСТ Р 34.10-2012, а также к атакам на его старую версию: ГОСТ Р 34.10-94, которые представлены в четвёртой главе.*

В третьей главе проанализированы наиболее распространённые в алгебре некоммутативные операции на предмет применения их в схеме открытого распределения ключей В.М.Сидельникова. Построены полиномиальные алгоритмы решения задачи факторизации относительно рассматриваемых

операций. Тем самым доказана нестойкость рассматриваемой схемы при использовании этих операций. Предложена новая некоммутативная операция с использованием символов степенного вычета, стойкая относительно рассмотренных атак.

В четвёртой главе предложена атака на старый стандарт цифровой подписи ГОСТ Р 34.10-1994. Универсальная подделка подписи в этом стандарте сведена к решению сравнения $x = R^x (\mathrm{mod} p)$ *относительно* $x \in \{0, \ldots, p-1\}$. *Получены новые оценки на число решений этого сравнения относительно пар* (R, x). *Доказательства оценок конструктивные, то есть позволяют строить указанные пары вероятностным полиномиальным алгоритмом. Это делается при помощи построения первообразных корней* $(\mathrm{mod} p)$, *наименьший положительный вычет которых взаимно прост с* $p-1$. *В некоторых случаях получены точные формулы, выражающие такие пары. Отметим здесь, что ГОСТ Р 34.10-1994 являлся основой для формирования последующих стандартов ГОСТ Р 34.10-2001 и ГОСТ Р 34.10-2012, для которых рассматриваемое сравнение модифицируется с использованием операции сложения точек на эллиптической кривой и может быть подвергнуто аналогичному анализу.*

Пятая глава посвящена исследованию алгоритма спуска Вейля для решения задачи дискретного логарифмирования в группе точек элиптической кривой. Изучены свойства представления Мамфорда дивизоров гиперэллиптических кривых. При помощи этого представления доказана теорема, характеризующая ядро спуска Вейля при использовании полей характеристики 2. Эта теорема выделяет случай, в котором образ гомоморфизма, называемого спуском Вейля, является нетривиальным и даже содержит ту же степень двойки, что и прообраз. Поэтому спуск Вейля может быть эффективно применён для решения задачи дискретного логарифмирования на гиперэллиптических кривых над полем характеристики 2.

В шестой главе получены новые блочные алгоритмы решения больших

разреженных систем линейных уравнений. Предложенные новые алгоритмы последовательного построения приближений Паде к произвольным формальным рядам по отрицательным и положительным степеням формальной переменной с матричными коэффициентами применены к построению ортогонального базиса пространства Крылова и решению соответствующей системы линейных уравнений. Идея использования блочных приближений для построения базиса пространства Крылова высказана и реализована впервые. Указанные алгоритмы демонстрируют возможность одновременного использования полезных свойств известных алгоритмов, основанных отдельно на последовательной ортогонализации пространства Крылова (алгоритм Монтгомери) и на построении приближений формальных рядов с матричными коэффициентами (алгоритм Видемана-Копперсмита). Данный подход имеет хорошие параллельные свойства, позволяющие отказаться от использования большого кластера при решении больших разреженных систем линейных уравнений. Поэтому, при анализе стойкости систем защиты информации, основанных на сложности задачи факторизации целых чисел, надо учитывать возможность применения нового метода на больших вычислительных ресурсах, связанных медленными каналами (Интернет).

Программа, реализующая алгоритм из шестой главы, которую можно использовать для атаки на схему RSA, внедрена в ОАО "Концерн "Автоматика", о чем имеется акт внедрения.

В седьмой главе техника 6 главы перенесена на случай большого конечного поля. Полученный алгоритм решения разреженных линейных систем над таким полем может быть применён к решению задачи дискретного логарифмирования в конечном поле.

Часть V

Приложение

Глава 1

Основные определения и теоремы в методе спуска Вейля

Основные понятия и определения

Доказательства сформулированных в этом подразделе утверждений можно найти в [48, гл.2, часть 1]. Близкие изложения — в [132, 133, 134]. Алгебраические свойства рассмотренных объектов можно посмотреть [135, 136, 137, 138, 139, 103]. Изложенные здесь определения и их свойства даны в единых обозначениях и необходимы для понимания основного результата пятой главы.

Точки

Пусть K - поле произвольной характеристики, $K(x)$ - поле рациональных функций от переменной x, F - алгебраическое расширение $K(x)$ конечной степени. В дальнейшем будет предполагаться, что поле K алгебраически замкнуто в F, т.е. каждый элемент поля F — алгебраический над K, принадлежит K. В этом предположении поле K будет называться полем констант F.

Подкольцо $\mathfrak{v}$ поля F называется кольцом нормирования, если оно удовлетворяет следующим условиям:

1) $\mathfrak{v}$ содержит поле K;

2) $\mathfrak{v}$ *не совпадает с* F*;*

3) если элемент z *из* F *не содержится в* $\mathfrak{v}$*, то* z^{-1} *содержится в* $\mathfrak{v}$*.*

Необратимые элементы кольца нормирования $\mathfrak{v}$ *образуют максимальный идеал* p *в кольце* $\mathfrak{v}$*.*

Точкой поля F *называется подмножество* p *поля* F*, являющееся идеалом некоторого кольца нормирования* $\mathfrak{v}$ *поля* F*, состоящим из всех необратимых элементов этого кольца.*

Кольцо $\mathfrak{v}$ *однозначно определяется заданием* p*. Факторкольцо* $\Sigma = \mathfrak{v}/p$ *есть поле. Это поле называется полем вычетов в точке* p*. Оно всегда является алгебраическим расширением поля* K *конечной степени. Степень этого расширения называется степенью точки и будет обозначаться* $d(p)$*.*

Будем считать, что символ ∞ *удовлетворяет обычным алгебраическим правилам. Точка* p *поля* F *задает отображение*

$$\psi : \quad F \longrightarrow \{\Sigma, \infty\}$$

поля F *в множество, состоящее из* Σ *и* ∞*, удовлетворяющее обычным правилам для гомоморфизмов*

$$\psi(a+b) = \psi(a) + \psi(b), \qquad \psi(ab) = \psi(a)\psi(b),$$

(если только выражения, стоящие в правых частях этих формул, определены) и такое, что $\psi(1) = 1$*. Элементы из* F*, которые не переводятся в* ∞*, называются конечными в этой точке, а остальные элементы называются бесконечными.*

Элементы поля F *можно рассматривать как функции на множестве точек этого поля. Поэтому мы будем иногда называть их функциями.*

Кольцо нормирования $\mathfrak{v}$ *точки* p *содержит такой элемент* t*, что* $p = t\mathfrak{v}$ *и* $\cap_{n=1}^{\infty} t^n\mathfrak{v} = \{0\}$*. Для каждого элемента* $a \in F$ *существует целое число* n *(оно может быть и отрицательным), что* $a \in t^n\mathfrak{v}$*. Наибольшее из таких чисел*

называется порядком элемента t *в точке* p *и обозначается* $\nu_p(a)$. *Для любых не равных нулю элементов* $a, b \in F$ *справедливы соотношения*

$$\nu_p(ab) = \nu_p(a) + \nu_p(b), \qquad \nu_p(a+b) \geq \min\{\nu_p(a), \nu_p(b)\}. \tag{1.1}$$

По определению полагается $\nu_p(0) = \infty$. *Функция* ν_p *не зависит от выбора элемента* t, *с условием* $p = t\mathfrak{v}$. *Так определенная функция называется функцией порядка в точке* p. *Элементы* t *поля* F, *для которых* $\nu_p(t) = 1$, *называются униформизирующими переменными, или локальными параметрами в точке* p. *Доказательства этих фактов см. в [140], гл.1, теорема 2.*

Пусть p *- точка поля* F. *Если элемент* $a \in F$ *принадлежит* p, *то говорят, что* p *является нулем элемента* a. *Если же* $a^{-1} \in p$, *то говорят, что* p *является полюсом* a. *Более того, при* $\nu_p(a) > 0$ *говорят, что* p *является нулем порядка* $\nu_p(a)$ *для элемента* a *и при* $\nu_p(a) < 0$ *— полюсом порядка* $-\nu_p(a)$.

Точки поля $K(x)$

Пусть $f(x)$ *- неприводимый многочлен кольца* $K(x)$ *и* $\mathfrak{v}_f$ *- множество элементов* $u = g/h$, *где* $g, h \in K[x]$, *причем* h *не делится на* f. *Множество* $\mathfrak{v}_f$ *есть кольцо нормирования поля* $K(x)$. *Обозначим* p_f *соответствующую точку. Ясно, что* p_f *состоит из всех элементов вида* $(fg)/h$, *где* g *и* h *— многочлены от* x, *причем* h *не делится на* f.

Еще один пример. Обозначим $\mathfrak{v}_\infty$ *множество элементов* $u = g/h$, *где* $g, h \in K[x]$, *причем* $\deg g \leq \deg h$. *Это множество также есть кольцо нормирования поля* $K(x)$. *Соответствующая ему точка состоит из всех элементов вида* g/h, *где* g *и* h *— многочлены от* x, *причем* $\deg g < \deg h$.

Заметим, что при $x' = x^{-1}$ *имеем* $K(x') = K(x)$. *Отсюда следует, что каждому неприводимому многочлену от* x' *с коэффициентами из* K *также соответствует некоторая точка поля* $K(x)$. *В частности,* $\mathfrak{v}_{x'} = \mathfrak{v}_\infty$.

Указанными примерами исчерпываются все точки поля $K(x)$.

Дивизоры

В отличие от книги [140] мы будем придерживаться аддитивной записи

дивизоров.

Каждой точке p поля F сопоставим целое число $n(p)$ так, чтобы $n(p) \neq 0$ только для конечного числа точек p. Полученная функция называется дивизором поля E и обозначается $D = \sum_p n(p)p$.

Если $p_1, \ldots, p_h$ — все точки поля F, входящие в дивизор D, т.е. все точки поля F, для которых $n_i = n(p_i) \neq 0$, то употребляется запись $D = \sum_{i=1}^{h} n_i p_i$. Дивизоры как функции на множестве точек поля F можно складывать, и тогда они образуют абелеву группу, которая обозначается $\mathrm{Div}(F)$.

Степенью дивизора называется сумма

$$\deg D = \sum_p d(p)n(p),$$

где $d(p)$ - степень точки p.

Дивизоры степени 0 образуют подгруппу в $\mathrm{Div}(F)$, которая обозначается $\mathrm{Div}^0(F)$. Дивизор называется целым, если для любой точки p поля F выполняется неравенство $n(p) \geq 0$. Для каждого элемента $a \in F, a \neq 0$ существует лишь конечное множество точек p, для которых $\nu_p(a) > 0$ (см. [140], гл.1, лемма 1).

Если $p_1, \ldots, p_h$ - все различные нули элемента $a \neq 0$ из поля F, и $n_i = \nu_{p_i}(a)$, то дивизор $\sum_{i=1}^{h} n_i p_i$ называется дивизором нулей элемента a.

Теорема 1.1. *Если элемент a из F не принадлежит K, то степень дивизора нулей этого элемента равна $[F : K(a)]$.*

Доказательство см. [140], гл.1, теорема 4.

Если эти результаты применить к элементу a^{-1} (вместо a), то получим, что элемент a имеет только конечное число полюсов. Если $q_1, \ldots, q_\ell$ - все различные полюса элемента a, и $m_i = -\nu_{q_i}(a)$, то целый дивизор $\sum_{i=1}^{\ell} m_i q_i$ называется дивизором полюсов элемента a. Так как $K(a^{-1}) = K(a)$, то справедливо

Следствие 1.1. *Если элемент a из F не принадлежит K, то степень дивизора полюсов этого элемента равна $[F : K(a)]$.*

Во введенных выше обозначениях дивизор

$$\sum_{i=1}^{h} n_i p_i - \sum_{i=1}^{\ell} m_i q_i$$

называется дивизором элемента a. Этот дивизор обозначается символом (a). Предыдущая теорема и ее следствие означают следующее утверждение.

Теорема 1.2. *Степень дивизора любого отличного от нуля элемента поля F равна 0.*

В силу (1.1) дивизоры вида (a) (главные дивизоры) образуют подгруппу $P(F)$ в группе $\mathrm{Div}^0(F)$ дивизоров нулевой степени. Факторгруппа группы $\mathrm{Div}^0(F)$ по подгруппе главных дивизоров $P(F)$ будет называться якобианом поля F и обозначаться $J(F)$.

Два дивизора будут называться эквивалентными друг другу, если их разность есть главный дивизор.

Квадратичные расширения поля $K(x)$

Будем предполагать теперь, что $[F : K(x)] = 2$, т.е. поле F есть квадратичное расширение поля рациональных функций $K(x)$. В этом случае найдется элемент $y \in F$ удовлетворяющий равенству

$$y^2 + f(x)y + h(x) = 0, \tag{1.2}$$

где $f(x), h(x) \in K[x]$ и такой, что $F = K(x, y)$. Многочлен $H(Y) = Y^2 + f(x)Y + h(x)$, очевидно, неприводим над полем $K(x)$.

В случае, если характеристика поля K равна 2, будем предполагать также, что $f(x) \neq 0$. Последнее условие обеспечивает отсутствие кратных корней у многочлена $H(Y)$. Оно также обеспечивает отсутствие кратных корней у минимального над $K(x)$ многочлена любого элемента поля F. Другими

словами, это свойство расширения $F \supset K(x)$ *называется сепарабельностью. Подробности см. [141], гл.7, п.4.*

Лемма 1.1. *Каждый идеал кольца* $K[x,y]$ *имеет* $K[x]$ *- базис вида* $(a, b+cy)$, *где* $a,b,c \in K[x]$, $a \notin K$, *причем*

1. многочлены a,b *делятся на* c, $\deg b < \deg a$ *и старшие коэффициенты многочленов* a,c *равны 1;*

2. справедливо сравнение

$$b^2 - fbc + hc^2 \equiv 0 \pmod{ac}.$$

Обозначим буквой $\mathcal{A}$ *подкольцо поля* F, *состоящее из элементов* F *целых над* $K[x]$ *(целое замыкание кольца* $K[x]$ *в* F *(см. [141], гл.9, п.1).*

Лемма 1.2. *Пусть характеристика поля* K *равна 2, многочлен* $f(x)$ *в (1.2) отличен от нуля. Тогда существует такой элемент* $\omega \in \mathcal{A}$, *что* $F = K(x,\omega)$ *и* $\mathcal{A} = K[x] + \omega K[x]$.

Умножение идеалов

Заменив образующую поля F *над* $K(x)$, *можно считать в силу леммы 1.2, что кольцо* $K[x,y]$ *есть целое замыкание* $K[x]$ *в* F *и следовательно целозамкнуто в поле* F.

Коммутативное кольцо без делителей нуля A *называется дедекиндовым, если оно удовлетворяет следующим трем условиям:*

1. A *нетерово, т.е. каждый идеал в* A *имеет конечный базис как* A *модуль.*

2. Каждый ненулевой простой идеал в A *максимален.*

3. A *целозамкнуто в своем поле частных.*

Теорема 1.3. *Кольцо* $\mathcal{A} = K[x,y]$ *дедекиндово.*

Теорема 1.4. *Любой ненулевой идеал дедекиндового кольца однозначно разлагается в произведение простых идеалов.*

Доказательство см. в [142], следствие 9.4.

Соответствие идеалов и дивизоров

Пусть $\mathfrak{p}$ *- простой идеал кольца* A, L *- поле частных* A. *Символом* $A_{\mathfrak{p}}$ *обозначается множество таких элементов* $a/s \in L$, *что* $a \in A, s \notin \mathfrak{p}$.

Теорема 1.5. *Пусть* A *– дедекиндово кольцо и* $\mathfrak{p} \subset A$ *– ненулевой простой идеал. Тогда* $A_{\mathfrak{p}}$ *есть кольцо нормирования. Соответствующей точкой* p *поля частных* L *служит совокупность элементов* a/s, $a \in \mathfrak{p}$, $s \notin \mathfrak{p}$, *т.е. множество* $p = \mathfrak{p}A_{\mathfrak{p}}$.

Доказательство см. в [142], теорема 9.3.

Теорема 1.6. *Пусть* A *- кольцо, содержащееся в поле* L. *Элемент* α *поля* L *тогда и только тогда цел над* A, *когда* α *лежит во всяком кольце нормирования поля* L, *содержащем* A.

См. [141], гл.13, предложение 16.

Пусть далее $A = \mathcal{A} = K[x, y]$ *и* $L = F$. *Теорема 1.5 сопоставляет каждому простому идеалу* $\mathfrak{p}$ *кольца* $K[x, y]$ *некоторую точку* $p = \mathfrak{p}A_{\mathfrak{p}}$ *поля* F. *Заметим, что при этом различным идеалам* $\mathfrak{p}_1$, $\mathfrak{p}_2$ *сопоставляются различные точки* p_1, p_2. *Действительно, если элемент* $a \in K[x, y]$ *таков, что* $a \in \mathfrak{p}_1$, $a \notin \mathfrak{p}_2$, *то* $a \in p_1$, $a \notin p_2$.

Лемма 1.3. *Если* $\mathfrak{v}$ *кольцо нормирования поля* F *и* $\mathfrak{m}$ *- соответствующая точка* F, *причем* $x \in \mathfrak{v}$, *то существует простой идеал* $\mathfrak{p} \subset \mathcal{A} = K[x, y]$ *такой, что* $\mathfrak{v} = \mathcal{A}_{\mathfrak{p}}$ *и* $\mathfrak{m} = \mathfrak{p}\mathcal{A}$.

Лемма 1.4. *Предположим, что многочлены* $f(x), h(x)$, *см. (1.2), удовлетворяют условиям*

$$\deg h = 2g + 1, \qquad \deg f(x) \leq g, \tag{1.3}$$

где g *- некоторое целое число. Тогда в поле* F *существует единственное кольцо нормирования (оно будет обозначаться* $\mathfrak{v}_\infty$*), не содержащее элемент* x.

Точка поля F, соответствующая кольцу нормирования $\mathfrak{v}_\infty$, будет обозначаться p_∞.

Пусть $\mathfrak{a}$ - нетривиальный идеал кольца $\mathcal{A} = K[x,y]$, целозамкнутого в поле $K(x,y)$. Согласно теореме 1.4 существует единственный набор простых идеалов $\mathfrak{p}_1, \ldots, \mathfrak{p}_r$ кольца $\mathcal{A}$ и единственный набор натуральных чисел $n_1, \ldots, n_r$ такие, что

$$\mathfrak{a} = \mathfrak{p}_1^{n_1} \cdots \mathfrak{p}_r^{n_r}. \tag{1.4}$$

Обозначим p_j точки поля F, соответствующие идеалам $\mathfrak{p}_j$, (см. теорему 1.5 и лемму 1.3.). В дальнейшем символом $div(\mathfrak{a})$ будет обозначаться дивизор

$$div(\mathfrak{a}) = \sum_{j=1}^{r} n_j p_j \in Div(F).$$

Ясно, что это обозначение зависит от выбора кольца $\mathcal{A} \subset F$. Однако выбрав и фиксировав это кольцо, мы сможем обозначать таким способом любой целый дивизор (т.е. $n_j \geq 0$) с условием, что все входящие в него точки конечны, т.е. содержат кольцо $\mathcal{A}$.

Если идеал $\mathfrak{a}$ имеет вид $\mathfrak{a} = \{a, y+b\}$ (см. лемму 1), то вместо $div(\mathfrak{a})$ будем также использовать обозначение $div(a,b)$.

Очевидно, равенство

$$div(\mathfrak{a}\mathfrak{b}) = div(\mathfrak{a}) + div(\mathfrak{b}),$$

т.е. отображение div есть изоморфизм полугруппы идеалов кольца $\mathcal{A}$ на полугруппу целых дивизоров, сосредоточенных в конечных точках.

Группа классов идеалов

Два нетривиальных (отличных от (0) и $\mathcal{A}$) идеала $\mathfrak{a}, \mathfrak{b}$ кольца $\mathcal{A}$ называются подобными (эквивалентными), если существуют ненулевые элементы $\alpha, \beta \in \mathcal{A}$, для которых выполняется равенство

$$(\alpha)\mathfrak{a} = (\beta)\mathfrak{b}. \tag{1.5}$$

Подобие есть отношение эквивалентности, и классы подобных идеалов образуют мультипликативную группу. Эта группа называется группой классов идеалов поля F. *Отношение подобия будет обозначаться* $\mathfrak{a} \ \mathfrak{s} \ \mathfrak{b}$, *а класс подобных идеалов, содержащий идеал* $\mathfrak{a}$, — *символом* $[\mathfrak{a}]$.

Якобиан и группа классов идеалов

Теорема 1.7. *Пусть кольцо* $K[x,y]$ *целозамкнуто в поле* $K(x,y)$ *и это поле обладает одной бесконечно удаленной точкой* p_∞ *степени 1. Тогда якобиан поля* F *изоморфен группе классов идеалов кольца* $\mathcal{A}$.

Этот изоморфизм задаётся следующим отображением. Сопоставим идеалу $\mathfrak{a}$ *дивизор* $D \in Div^0(F)$ *следующим образом:*

$$\phi : \mathfrak{a} \longrightarrow D = \sum_{j=1}^{r} n_j p_j - n p_\infty = div(\mathfrak{a}) - deg\mathfrak{a} \cdot p_\infty,$$

где $n = n_1 + \cdots + n_r$.

Конорма и норма

Пусть $E \subset F$ – *два поля алгебраических функций. Их поля констант обозначаются соответственно* L *и* K. *При этом будет предполагаться, что* $L = K \cap E$. *Ниже мы установим некоторые соответствия между дивизорами полей* E *и* F.

Говорят, что точка P *поля* F *лежит над точкой* p *поля* E, *если* $p \subset P$. *Говорят также, что* P *является* продолжением p, *а* p *является* ограничением P *на* E, *и это обозначается* $P|p$.

Лемма 1.5. *Во введенных выше условиях и обозначениях эквивалентны следующие утверждения:*

1. $P|p$;

2. $\mathfrak{v}_p \subset \mathfrak{v}_P$;

3. существует целое $e \geq 1$ *такое, что* $\nu_P(\alpha) = e\nu_p(\alpha)$ *для всех* $\alpha \in E$.

Более того, если $P|p$, *то*

$$p = P \cap E, \qquad \mathfrak{v}_p = \mathfrak{v}_P \cap E.$$

Доказательство. *См. [143], гл. 3, предложение III.1.4.*

Целое число e *из формулировки леммы называется* индексом ветвления P *над* p. *Этот индекс обозначается* $e(P|p)$. *Из леммы 1.5 следует, что существует вложение полей вычетов* $E_p = \mathfrak{v}_p/p \longrightarrow F_P = \mathfrak{v}_P/P$, *при котором* $\alpha \pmod p$ *переводится в* $\alpha \pmod P$. *Расширение* $F_P \supset E_p$ *является, очевидно, алгебраическим, и его степень* $f(P|p) = [F_P : E_p]$ *называется* относительной степенью *точки* P *над* p.

Лемма 1.6. *1. Для каждой точки* P *поля* F *существует единственная точка* p *поля* E *с условием* $P|p$, *а именно* $p = P \cap E$.
2. Обратно, для любой точки p *поля* E *существует по крайней мере одна, продолжающая ее точка* P *поля* F. *Множество таких продолжений конечно.*

Доказательство. *См. [143], гл. 3, Предложение III.1.7.*

Определим теперь отображение конормы из группы дивизоров $Div\ (E)$ *поля* E *в группу* $Div\ (F)$ *дивизоров поля* F.

Для каждой точки p *поля* E *определим ее* конорму *равенством*

$$Con_{F/E}(p) = \sum_{P|p} e(P|p) \cdot P,$$

где суммирование ведется по всем точкам P *поля* F, *продолжающим точку* p. *Для каждого дивизора* $D = \sum_p n(p)p \in Div\ (E)$ *определим его конорму равенством*

$$Con_{F/E}(D) = \sum_p n(p) Con_{F/E}(p).$$

Лемма 1.7. *Если* a *– ненулевой элемент поля* E *и* $(a)^E \in Div\ (E)$, $(a)^F \in Div\ (F)$ *– главные дивизоры, определенные элементом* a *в полях* E *и* F *соответственно, то*

$$Con_{F/E}((a)^E) = (a)^F.$$

Доказательство. *См. [143], гл. 3, Лемма III.1.10.*

В частности, эта лемма означает, что образом главного дивизора при отображении конормы вляется главный дивизор, и потому это отображение может быть определено на группе классов дивизоров.

Алгоритм разложения дивизора по базе.

Необходимые определения.

Пусть k *– поле из* $q = 2^r$ *элементов,* $u(x)$ *и* $v(x)$ *– многочлены над полем* $k, degv(x) \leq g, degv(x) = 2g + 1$, *где* g *– натуральное число, большее* 1. *Тогда уравнение*

$$y^2 + u(x)y + v(x) = 0 \tag{1.6}$$

задает гиперэллиптическую кривую рода g, *определенную над полем* k.

Назовем базой дивизоров произвольное непустое множество B, *состоящее из пар многочленов* $(a(x), b(x))$, *определенных над полем* k *и обладающих свойствами:*

$$\deg b(x) < \deg a(x) \leq g \tag{1.7}$$

$$b^2 + ub + v = 0 \ (moda), \tag{1.8}$$

и многочлен $a(x)$ *неприводим над полем* k.

Как было показано в главе 5, эти пары взаимнооднозначно соответствуют классам дивизоров с представителями $div(a, b)$.

Постановка задачи.

Пусть дан дивизор $D = (a, b)$ *и требуется разложить его по базе* $B = \{D_i\}$ *(*D_i *– дивизор,* i *пробегает некоторое множество индексов), или установить, что это невозможно.*

Описание алгоритма.

Шаг 1. Проверить, раскладывается ли многочлен a *на множители* a_i, *где* a_i *– первые компоненты дивизоров* D_i *из базы* B. *Если разложить не удается,*

то это значит, что данный дивизор не раскладывается по базе. Если найдено представление $a = \prod_{i=1}^{s} a_i^{k_i}$, *идем на шаг 2.*

Шаг 2. Имеет место разложение $D = \sum_{i=1}^{s} \varepsilon_i k_i D_i$, *где* $\varepsilon_i \in \{-1, +1\}$. *Остается определить знаки:* $\varepsilon_i = 1$, *если* $b = b_i \pmod{a_i}$, *и* $\varepsilon_i = -1$, *если* $b = b_i + u \pmod{a_i}$.

Обоснование алгоритма.

Если a *не раскладывается в произведение* a_i, *значит, существует* $x_0 \in \overline{F}_q$ *такой, что* $a(x_0) = 0, a_i(x_0) \neq 0$ *для всех* i. *То есть точка* $P(x_0, y_0)$, *принадлежащая спектру дивизора* D, *не принадлежит спектрам дивизоров из базы* B, *и поэтому его разложение по этой базе невозможно.*

Пусть теперь разложение $a(x)$ *на множители в шаге 1 найдено. Уравнение* $t^2 + u(x)t + v(x) = 0$ *в поле* $k[x]/a_i(x)$ *имеет только два, возможно, одинаковых решения:* $b_i, b_i + u$, *поэтому* $b(x) \equiv b_i(x) (\mathrm{mod}(a_i(x)))$, *или* $b(x) \equiv b_i(x) + u(x) (\mathrm{mod}(a_i(x)))$. *Если точка* $P(x_0, y_0)$ *принадлежит спектру дивизора* D, *то* $a(x_0) = 0, b(x_0) = y_0$, *и для некоторого* i *выполнено:* $a_i(x_0) = 0, b_i(x_0) = y_0 или b_i(x_0) = y_0 + u(x_0)$. *Теперь* $b_i(x_0) = y_0 \Leftrightarrow b_i(x_0) = b(x_0)$. *Поскольку это должно быть выполнено для любого корня* x_0 *многочлена* a_i, *то* $a_i | b - b_i$.

Построение базы дивизоров.

База дивизоров строится следующим образом. Выбираются все неприводимые многочлены над полем k *степени не выше* c. *Параметр* c *влияет на размеры матриц, которые нужно будет приводить, поэтому он может быть выбран эмпирически, исходя из характеристик вычислительной техники и времени работы алгоритмов. Чтобы гарантировать разложение теоретически, этот параметр надо выбирать так:*

$$c = \lceil \log_q(L(\rho)) \rceil, \quad L(\rho) = exp(\rho \sqrt{g \ln(q) \ln(g \ln(q))}), \tag{1.9}$$

$$\rho = \min\left(\frac{\sqrt{2}}{4};\ \frac{1}{2\sqrt{\theta}}\left(\sqrt{\frac{\theta+3}{3}}-1\right)\right),\quad \theta = \frac{g}{\ln q}.$$

Далее для каждого неприводимого многочлена $a(x)$ *находим многочлен* $b(x)$, *удовлетворяющий сравнению (1.8) (алгоритм проверки описан ниже). Если многочлен* $b(x)$ *существует, то описанный ниже алгоритм его находит и дивизор* $D = (a, b)$ *добавляется в базу.*

Алгоритм решения сравнения (1.8) .

Пусть $k = \frac{\mathbb{Z}_2[t]}{f(t)}$, *где* $\deg f(t) = r$. *Обозначим* $L = \frac{k[x]}{a(x)}$, *где многочлен* $a(x)$ *– неприводимый степени* n. *Тогда* L *– конечное поле из* 2^n *элементов. Фактически нам надо теперь решить уравнение (1.8) в этом поле. Заметим, что если* $a(x)$ *не делит* $u(x)$, *то, домножая (1.8) на* $(u^{-1})^2$, *сведем (1.8) к виду*

$$z^2 + z + w = 0 \,(\mathrm{mod}\, a), \tag{1.10}$$

где $z = bu^{-1}$, $w = vu^{-2}$.

Неизвестный элемент z *поля* L *можно разложить по базису* $1, x, \ldots, x^{n-1}$, *а затем каждый коэффициент в полученном разложении рассмотреть как элемент поля* k *и разложить по базису* $1, t, \ldots, t^{r-1}$. *В результате получим:*

$$z = \sum_{i,j} z_{ij} t^i x^j, \tag{1.11}$$

где в правой части ведется суммирование по $i = 0, \ldots, r-1$ *и* $j = 0, \ldots, n-1$, *а* $z_{ij} \in \mathbb{Z}_2$.

Введем в рассмотрение набор величин $\alpha_{ij}^l \in \mathbb{Z}_2$, $i = 0, \ldots, r-1$, $j = 0, \ldots, n-1$, $l = 0, \ldots, 2n-2$, *определяемые разложением (аналогичным разложению (1.11)) степеней* x^l *по базису* $t^i x^j$ *:*

$$x^l = \sum_{i,j} \alpha_{ij}^l t^i x^j, \quad l = 0, \ldots, 2n-2. \tag{1.12}$$

Суммирование ведётся по $i = 0, \ldots, r-1$, $j = 0, \ldots, n-1$.

Набор $\beta_s^k \in \mathbb{Z}_2$ *определим соотношениями:*

$$t^k = \sum_s \beta_s^k t^s, \quad k = 0, \ldots, 3(r-1), \tag{1.13}$$

где суммирование в правой части идет по $s = 0, \ldots, r-1$.

Заметим, что наборы $\{\alpha_{ij}^l\}$ *и* $\{\beta_s^k\}$ *можно считать известными, т.к. они могут быть эффективно вычислены на основе известных многочленов* $f(t)$ *и* $a(x)$. *На самом деле, этап набора базы дивизоров занимает очень малое время в сравнении с набором соотношений, поэтому какое-либо оптимизирование этого процесса не имеет большого практического смысла.*

Теперь вернемся к уравнению (1.10). Перенесем w *в правую часть, тогда левая часть будет линейна по* z. *Разложим* z *и* w *по базису, как это сделано в (1.11), затем воспользуемся наборами чисел (1.12) и (1.13) для разложения старших степеней* x *и* t *по базису. Тогда мы сведем уравнение (1.10) к эквивалентной системе линейных уравнений с* nr *неизвестными* z_{ij}, *решение которой нужно искать в поле* $\mathbb{Z}_2$, *что уже не представляет сложностей. Проделаем выкладки:*

$$z^2 + z = \sum_{i,j}((z_{ij}t^i x^j)^2 + z_{ij}t^i x^j) = \sum_{i,j}(z_{ij}t^{2i}x^{2j} + z_{ij}t^i x^j)$$

$$= \sum_{i,j}(z_{ij}t^{2i}\alpha_{ks}^{2j}t^k x^s + z_{ij}t^i x^j) = \sum_{i,j}\sum_{k,s}(z_{ij}\alpha_{ks}^{2j}x^s t^{2i+k} + z_{ij}t^i x^j)$$

$$= \sum_{i,j}\sum_{k,s}(z_{ij}\alpha_{ks}^{2j}x^s \sum_l(\beta_l^{2i+k}t^l) + z_{ij}t^i x^j)$$

$$= \sum_{i,j}\sum_{k,s}\sum_l z_{ij}(\alpha_{ks}^{2j}\beta_l^{2i+k}t^l x^s + z_{ls}t^l x^s).$$

Здесь i, k, l *изменяются в интервале* $[0, r-1]$, *а* j, s *— в интервале* $[0, n-1]$. *Теперь, считая, что* $w = \sum_{l,s} w_{ls}t^l x^s$, *и приравнивая коэффициенты при одинаковых членах* $t^l x^s$, *получаем систему линейных уравнений, равносильную*

сравнению 1.8:

$$\sum_{i,j,k} z_{ls} + z_{ij}\alpha_{ks}^{2j}\beta_l^{2i+k} = w_{ls}, \quad l = 0, \ldots, r-1, \quad s = 0, \ldots, n-1 \tag{1.14}$$

относительно неизвестных z_{ls}. *Находим какое-либо её решение* u, *если оно существует, восстанавливаем вторую координату дивизора* $D = (a, b)$.

В случае, когда мы не можем разделить уравнение (1.8) на u, *то есть* a *делит* u, *оно сводится к виду*

$$b^2 + v = 0 (\mathrm{mod}\, a). \tag{1.15}$$

Это уравнение всегда имеет решение в поле из 2^n *элементов. Это решение равно* $v^{2^{n-1}}$. *Его можно получить возведением* v *в квадрат* $n-1$ *раз по модулю* a.

Глава 2

Алгоритм Ланцоша-Монтгомери

Алгоритм Монтгомери строит последовательность блоков V_i *из* n *(длина машинного слова) векторов, принадлежащих* $\mathbb{F}^N$, *и последовательность блоков* W_i *из* $n_i \leq n$, *векторов из* $\mathbb{F}^N$, *порождающих подпространства, удовлетворяющие условию теоремы 6.4.*

На i*-м шаге алгоритма имеется блок* V_i, *A-ортогональный всем* $W_j, 0 \leq j < i$. *В начале алгоритма* V_0 *выбирается равным правой части системы (6.2). Блок* W_i *строится из блока* V_i *при помощи "Вспомогательного алгоритма"(см. далее) и состоит из столбцов блока* V_i, *возможно не всех, так, что*

$$W_i = V_i S_i,$$

где $S_i \in \mathbb{F}^{n \times n_i}$ *такова, что каждый её столбец состоит из нулей и ровно одной единицы, причём*

$$(I_n - S_i {S_i}^T)(I_n - S_{i+1} {S_{i+1}}^T) = 0. \tag{2.1}$$

Ввиду такой структуры ${S_i}^T S_i = I_{n_i}$, *а* $S_i {S_i}^T$ *является диагональной матрицей размера* N *с единицами на диагонали в тех местах, которые соответствуют отобранным в* W_i *столбцам. Помимо матрицы* S_i, *"Вспомогательный алгоритм"выдаёт матрицу*

$$W_i^{inv} = S_i(W_i{}^T A W_i)^{-1} S_i{}^T = S_i(S_i{}^T V_i{}^T A V_i S_i)^{-1} S_i{}^T. \tag{2.2}$$

Затем вычисляется V_{i+1} *по формуле:*

$$V_{i+1} = A V_i S_i S_i{}^T + V_i - \sum_{j=0}^{i} W_j C_{i+1,j}, \tag{2.3}$$

где $C_{i+1,j}$ *выбраны так, чтобы*

$$W_j{}^T A V_{i+1} = 0, 0 \le j \le i, \tag{2.4}$$

а именно

$$C_{i+1,j} = (W_j{}^T A W_j)^{-1} W_j{}^T A (A W_i S_i{}^T + V_i). \tag{2.5}$$

Действительно, применяя индукцию по i*, находим*

$$W_j{}^T A V_{i+1} = W_j{}^T A (A W_i S_i{}^T + V_i) - \sum_{k=0}^{i} W_j{}^T A W_k C_{i+1,k} =$$

$$W_j{}^T A (A W_i S_i{}^T + V_i) - W_j{}^T A W_j C_{i+1,j} = 0.$$

Из формулы (2.5) при $j < i$ *теперь получим*

$$C_{i+1,j} = (W_j{}^T A W_j)^{-1} W_j{}^T A^2 W_i S_i{}^T,$$

а по формуле (2.3)

$$W_j{}^T A^2 W_i = S_j{}^T S_j W_j{}^T A^2 W_i = S_j{}^T (A W_j S_j{}^T)^T A W_i =$$

$$= S_j{}^T (V_{j+1} - V_j + \sum_{k=0}^{j} W_k C_{j+1,k})^T A W_i =$$

$$S_j{}^T V_{j+1}{}^T A W_i - W_j{}^T A W_i = S_j{}^T V_{j+1}{}^T A W_i.$$

Поэтому при $j < i$

$$C_{i+1,j} = (W_j{}^T A W_j)^{-1} S_j{}^T V_{j+1}{}^T A W_i S_i{}^T. \tag{2.6}$$

Используя свойство (6.7), формулу (2.3) можно упростить. Перепишем эту формулу в виде:

$$V_{i+1} = (A + I_n) V_i S_i S_i{}^T + V_i (I_n - S_i S_i{}^T) - \sum_{j=0}^{i} W_j C_{i+1,j}.$$

Рассматривая только столбцы, соответствующие единицам в матрице $I_n - S_i S_i{}^T$, *получим*

$$\langle V_i (I_n - S_i S_i{}^T) \rangle \subseteq \langle V_{i+1} \rangle + \langle W_i \rangle + \cdots + \langle W_0 \rangle,$$

откуда, поскольку по построению $\langle V_k S_k S_k{}^T \rangle = \langle W_k \rangle$ *для любого* k, *а* $(I_n - S_i S_i{}^T) S_{i+1} S_{i+1}{}^T = I_n - S_i S_i{}^T$, *домножая на* $S_{i+1} S_{i+1}{}^T$, *получим*

$$\langle V_i (I_n - S_i S_i{}^T) \rangle \subseteq \langle W_{i+1} \rangle + \langle W_i \rangle + \cdots + \langle W_0 \rangle, \langle V_i S_i S_i{}^T \rangle = \langle W_i \rangle.$$

Построенные в данном алгоритме блоки удовлетворяют следующему свойству:

Лемма 2.1. $\langle V_i \rangle \subseteq \langle W_{i+1} \rangle + \langle W_i \rangle + \cdots + \langle W_0 \rangle$.

Применяя эту лемму из формулы (2.6), получим, что $C_{i+1,j} = 0$ *при* $j < i - 2$. *Теперь формулу (2.3), применяя (2.6), (2.2), можно переписать в виде*

$$V_{i+1} = A V_i S_i S_i{}^T + V_i - V_i W_i{}^{inv} V_i{}^T A (A V_i S_i S_i{}^T + V_i) -$$
$$V_{i-1} W_{i-1}{}^{inv} V_{i-1}{}^T A^2 V_i S_i S_i{}^T - V_{i-2} W_{i-2}{}^{inv} V_{i-2}{}^T A^2 V_i S_i S_i{}^T.$$

Для упрощения процедуры вычисления по этой формуле выразим скалярные произведения $V_{i-1}{}^T A^2 V_i, V_{i-2}{}^T A^2 V_i$ *через скалярные произведения вида* $V_i{}^T A V_i, V_i{}^T A^2 V_i$. *Применяя формулу (2.3) и свойство (2.4), получим*

$$S_{i-1}{}^T V_{i-1}{}^T A^2 V_i = (AW_{i-1})^T AV_i = S_{i-1}{}^T V_i{}^T AV_i,$$

$$S_{i-2}{}^T V_{i-2}{}^T A^2 V_i = (AW_{i-2})^T AV_i = S_{i-2}{}^T V_{i-1}{}^T AV_i =$$

$$= S_{i-2}{}^T V_{i-1}{}^T A(AW_{i-1}S_{i-1}{}^T + V_{i-1} - W_{i-1}C_{i,i-1}) =$$

$$= S_{i-2}{}^T V_{i-1}{}^T A(I_N - V_{i-1}W_{i-1}{}^{inv}V_{i-1}{}^T A)(AW_{i-1}S_{i-1}{}^T + V_{i-1}) =$$

$$= S_{i-2}{}^T (I_n - V_{i-1}{}^T AV_{i-1}W_{i-1}{}^{inv})(V_{i-1}{}^T A^2 V_{i-1}S_{i-1}S_{i-1}{}^T + V_{i-1}{}^T AV_{i-1}).$$

Таким образом,

$$V_{i+1} = AV_iS_iS_i{}^T + V_iD_{i+1} + V_{i-1}E_{i+1} + V_{i-2}F_{i+1},$$

где при $i \geq 0$

$$D_{i+1} = I_n - W_i^{inv}(V_i{}^T A^2 V_iS_iS_i{}^T + V_i{}^T AV_i),$$

$$E_{i+1} = -W_{i-1}^{inv}V_i{}^T AV_iS_iS_i{}^T,$$

$$F_{i+1} = -W_{i-2}^{inv}(I_n - V_{i-1}{}^T AV_{i-1}W_{i-1}^{inv})(V_{i-1}{}^T A^2 V_{i-1}S_{i-1}S_{i-1}{}^T + V_{i-1}{}^T AV_{i-1})S_iS_i{}^T.$$

Здесь $W_j{}^{inv} = V_j = 0, S_j = I_n$ *при* $j < 0$.

Допущение: *Таким образом, указанные процедуры строят нужную последовательность блоков только при выполнении условия (2.7), а именно:*

$$(I_n - S_iS_i{}^T)(I_n - S_{i+1}S_{i+1}{}^T) = 0_n, i \geq 1. \tag{2.7}$$

В противном случае нарушается условие A*-ортогональности блоков* W_i*. Это происходит, например, в случае, когда один из вектор-столбцов блока* V_i A*-ортогонален всем вектор-столбцам блоков* V_i, V_{i+1}.

Алгоритм заканчивает свою работу при условии $V_i{}^T AV_i = 0$*, при некотором* $i = m$.

Если при этом $V_m = 0$*, то, очевидно, выполняются все условия теоремы, и мы находим решение по формуле (6.7). Если же* $V_m \neq 0$*, то для завершения*

алгоритма необходимо провести дополнительные вычисления, которые мы опишем чуть позже.

Завершение алгоритма Монтгомери.

Пусть алгоритм Монтгомери для матрицы A и правой части $B = AY$ завершился с параметрами X, V_m.

Как указано выше, если $V_m = 0$, то X является решением системы (6.2). Рассмотрим случай, когда $V_m \neq 0$. В этом случае пространство $\langle V_m \rangle$ A-ортогонально пространству $W = \langle W_0 \rangle + \cdots + \langle W_{m-1} \rangle$, а также A-ортогонально себе. На практике размерность $\langle V_m \rangle$ небольшая (приблизительно 2). **Допущение:** *предположим, что $\langle V_m \rangle$ содержит все вектора пространства Крылова $V = \langle V_0 \rangle + \langle AV_0 \rangle + \dots$, ортогональные всему этому пространству. Тогда AV_m имеет вид $w_0 + w_1 + \cdots + w_m, \langle w_i \rangle \subseteq \langle W_i \rangle, i = 1, 2, \dots, m-1; \langle w_m \rangle \subseteq \langle V_m \rangle$. Получаем при $j \in \{1, \dots, m-1\}$ цепочку равенств*

$$w_j{}^T A W_j = (w_0 + w_1 + \cdots + w_m)^T A W_j = (AV_m)^T A W_j =$$
$$(V_m)^T A (AW_j) \subseteq (V_m)^T A\, V = \{0\},$$

которая, ввиду невырожденности A-скалярного произведения на W_j, показывает, что $w_j = 0$ при $j = 0, 1, \dots, m-1$, то есть $AV_m \subseteq \langle V_m \rangle$. Из формулы (6.7), как и раньше, получим, что вектора блока $AX - B$ A-ортогональны W, то есть $\langle AX - B \rangle \subseteq \langle V_m \rangle$. Значит, линейный оператор A отображает сумму пространств $\langle X - Y \rangle + \langle V_m \rangle$ в $\langle V_m \rangle$. Если $\langle Y \rangle \not\subseteq V$, то размерность первого пространства больше, чем размерность второго, и при помощи Гауссовых исключений в равенстве $A(X - Y \| V_m) = Z, \langle Z \rangle \subseteq \langle V_m \rangle$ можно найти элемент из ядра A.

Описанную процедуру можно применить сразу для получения решения исходной системы (6.1). Образуем двойной блок $Z = (X - Y \| V_m) \in \mathbb{F}^{N \times 2n}$ путем конкатенации блоков $X - Y$ и V_m (при этом допускается $V_m = 0$). Вычислим двойной блок DZ и элементарными преобразованиями U над столбцами

приведем его к треугольному виду. При этом ненулевые стобцы двойного блока ZU, *соответствующие нулевым столбцам треугольной матрицы* DZU, *если такие найдутся, дадут решения несимметричной системы (6.1).*

Блочный фактор.

Заменим в предыдущих рассуждениях размер матриц, равный n *на* ns, *а параметр* s *назовём блочным фактором. Проследим, как изменится время работы частей алгоритма Монтгомери и используемая при этом оперативная память.*

Прежде всего, число шагов алгоритма сократится с $\frac{N}{ns}$ *до* $\frac{N}{ns}$. *Кроме того, вычиление элементов* $\alpha_i = B^T A^i B$ *можно проводить на независимых вычислителях, разделив правый множитель* B *на* s *столбцов. Однако это не гарантирует уменьшение общего времени работы. Алгоритм построчного умножения матриц из* $\mathbb{F}^{ns\times ns}$ *будет иметь сложность* n^2s^3. *Действительно, последовательно проходя каждую из* ns *строк левой матрицы, мы для каждой встречающейся единицы должны осуществить сложение* s *машинных слов. При этом количество единиц в строке оценивается величиной* ns. *Аналогичные рассуждения показывают, что и трудоёмкость умножения плотных блоков, то есть случайно заполненных нулями и единицами блоков из* $\mathbb{F}^{tns\times ns}$ *друг на друга с ростом* s *также растёт, как* s^3. *Отметим, однако, что этот вывод может быть скорректирован в сторону уменьшения но не более, чем до* s^2 *для вычислительных машин с процессорами, выполняюшими за один такт несколько (*s*) однотипных операций с машинными словами. Таким образом, с ростом параметра* s *время на выполнение одних частей алгоритма уменьшается, а на выполнение других - увеличивается. Возникает задача определения оптимального значения блочного фактора, к которой мы вернёмся несколько позже. Теперь же отметим ещё, что объём оперативной памяти для хранения текущих блоков пространства Крылова увеличивается в* s *раз, а на хранение текущих многочленов с матричными коэффициентами - также*

в s раз, принимая во внимание, что максимальная степень рассматриваемых многочленов в s раз меньше прежнего значения $\frac{N}{n}$.

Глава 3

Алгоритм Видемана-Копперсмита

Алгоритм ищет ненулевое решение системы $Ax = 0$ *для большой разреженной матрицы* A *над конечным полем* $\mathbb{F} = GF(2), A \in \mathbb{F}^{N\times N}, x \in \mathbb{F}^{N\times n}$. *Здесь* n - *длина машинного слова или кратное ей число.*

Вход: $A \in \mathbb{F}^{N\times N}; \Delta, m', n' \in \mathbb{N}, m \geq n$. *Полагаем, что* m', n' *кратны длине машинного слова.*

Шаг 1: Выбираем случайные матрицы $X \in \mathbb{F}^{m\times N}, Y \in \mathbb{F}^{N\times n'}$.

Шаг 2: Положим $\delta' = [\frac{N}{m'}], \delta'' = [\frac{N}{n'}]$. *Вычисляем последовательно матрицы* $H_i = XA^iY \in \mathbb{F}^{m'\times n'}; i = 0, \ldots, \delta' + \delta'' + \Delta$.

Шаг 3: Вычисляем блоки $g_j \in \mathbb{F}^{n'\times n}, 0 \leq j \leq d, d \leq \delta''$ *такие, что*

$$H_i g_0 + H_{i+1} g_1 + \cdots + H_{i+d} g_d = 0, 0 \leq i \leq \delta' + \Delta,$$

или

$$XA^i\omega = 0, \quad \text{где} \quad \omega = \sum_{j=0}^{d} A^j Y g_j.$$

Эти коэффициенты ищутся в алгоритме как коэффициенты приближения ряда $\sum_{i=0}^{\infty} H_i \lambda^i$.

Шаг 4: Вычисляем ω. *Если* $\omega \neq 0$, *вычисляем наименьшее неотрицательное* f *такое, что* $A^f\omega = 0$.

Выход: $x = A^{f-1}\omega$.

Глава 4

Реализация последовательной версии нового алгоритма решения разреженных систем линейных уравнений

В настоящем документе представлены результаты работы по реализации последовательной (не распараллеленной) версии алгоритма Черепнёва.

Постановка задачи и обозначения.

Пусть $C \in \mathbb{F}_2^{M \times N}$, *где* $M, N \in \mathbb{N}, M < N$. *Требуется найти решение* $X \in \mathbb{F}_2^{N \times n}$ *однородной системы* $CX = 0$, *где* $n \in \mathbb{N}$ – *блочный фактор.*

В настоящей реализации принято значение блочного фактора $n = 64$.

Программные средства.

Для реализации последовательной версии алгоритма Черепнёва были созданы следующие программы:

- *Библиотека для работы с основными объектами*
- *Генератор тестовых матриц*
- *Программа-реализация алгоритма*
- *Программа проверки полноты построенного пространства*

Все программы написаны на языке Си и могут быть собраны на ОС Windows или ОС Linux.

Библиотека для работы с основными объектами.

Библиотека для работы с основными объектами соджержит реализацию всех операций над базовыми объектами, используемыми в алгоритме Черепнёва; все остальные программы используют ее вызовы.

Основные операции, реализованные в библиотеке:

- `mat64x64` – *матрицы из* $\mathbb{F}_2^{n\times n}$ *(„маленькие“ матрицы),*
  - *операции сложения, умножения, сравнения, приведения к треугольному виду и вычисления ранга, обращения по Монтгомери и др.*
- `matNx64` – *матрицы из* $\mathbb{F}_2^{N\times n}$ *(„средние“ матрицы или матрицы-“вектора“),*
  - *операции сложения, скалярного произведения, сравнения, приведения к треугольному виду и вычисления ранга, умножение на „маленькие“ матрицы и др.*
- `poly64x64` – *многочлены из* $\mathbb{F}_2^{n\times n}[\lambda]$ *(многочлены над „маленькими“ матрицами)*
  - *операции сложения, скалярного произведения (на ряд), сравнения, приведения к треугольному виду и вычисления ранга, умножение на „маленькие“ матрицы и на* λ *и др.*
- `matNx6N` – *разреженные матрицы из* $\mathbb{F}_2^{N\times N}$ *(„большие“ матрицы)*
  - *операции порождения случайных матриц с заданыыми параметрами, сохранение или восстановление из файла, умножение на матрицы-*

“вектора“, двойное (симметричное) умножение на матрицы-“вектора“ и др.

Программа-реализация алгоритма.

Программа `algo` *реализует последовательный вариант алгоритма Череннёва.*

На вход программе подается файл, содержащий разреженную матрицу, в формате, генерируемом программой `genmtx`*. По ходу работы на экран выводится информация о подпространствах на текущем шаге, а также информация о проверке решения. В результате работы на стандартный поток ошибок выводится матрица-“вектор"из ядра входной матрицы. Также порождается бинарный файл* `vec.bin`*, содержащий все подпространства пространства Крылова, построенные алгоритмом.*

Формат вызова программы.

```
algo <matrix>
```

Здесь параметр `matrix` *– имя файла, содержащего тестовую разреженную матрицу.*

Примерный вывод программы.

```
Seeding with 0x21356B1E
Starting with parameters: N = 1010, n = 64, max iterations = 79, N/n = 15.
Step 1/15 (64/0)
Step 2/15 (63/0)
Step 3/15 (64/1)
Step 4/15 (64/0)
Step 5/15 (63/0)
Step 6/15 (63/1)
Step 7/15 (64/1)
Step 8/15 (64/0)
```

```
Step 9/15 (63/0)
Step 10/15 (64/1)
Step 11/15 (63/0)
Step 12/15 (63/1)
Step 13/15 (64/1)
Step 14/15 (62/0)
Step 15/15 (63/2)
Step 16/15 (38/1)
 Q is the denominator!
 rk W_0+...+W_m-1 = 998
 rk U = 2
 rk V_m = 2
 rk U+V_m = 2
 rk X+Y = 12
 rk CtC(U+V_m) = 0
 rk CtC(X+Y) = 2
 rk CtC(U+V_m|X+Y) = 2
 -----------
 rk X' = 10
  (Verified CtCX'=0 OK!)
 -----------
 rk CX' = 0
 rk X = 10
  (Verified CX=0 OK!)
Done in 0 sec.
```

На каждом шаге работы в скобках выводится размерность нового построенного подпространства (через косую – количество использованых векторов с предыдущего шага). В конце указываются ранги: векторов,

не использованных на предыдущих шагах, (U); последнего построенного подпространства (V_m); их комбинации; ранг решения системы $CtCX' = 0$ (с проверкой); ранг решения системы $CX = 0$ (с проверкой).

Результаты испытаний.

Ниже в таблице представлены результаты испытаний, проведенных на компьютере со следующими характеристиками: Intel Core 2 Duo E8550, 4Гб, ОС Linux.

Все матрицы имели по 11 ненулевых элементов в столбце.

Тест	N	M	$rkW_1 + \cdots + W_{m-1}$	$rkU + V_m$	rkX	*Время, с*
1	1010	1000	998	2	10	0
2	1010	1000	998	2	10	1
3	1010	1000	996	4	10	1
4	2010	2000	1998	1	11	1
5	2010	2000	1998	2	10	1
6	2010	2000	1996	4	10	1
7	5010	5000	5000	0	10	5
8	5010	5000	4998	2	10	5
9	5010	5000	4998	2	10	5
10	10010	10000	9997	3	10	18
11	10010	10000	9997	3	10	18
11	10010	10000	10000	0	10	18
12	50010	50000	49998	1	11	431
13	50010	50000	49998	1	10	430
14	50010	50000	49996	1	13	430

Литература

[1] Diffie W., Hellman M. New directions in cryptography// IEEE Transactions and Information Theory.-1976.-v.IT-22-p.637-647

[2] Нестеренко Ю.В. Теория чисел.-М.:Издательский дом "Академия 2008.-272с.

[3] Stewart I.N., Tall D.O. Algebraic number theory.-New York-London,Chapman and Hall,1986.

[4] Хассе Г. Лекции по теории чисел.-М.: ИЛ,1953.-528с.

[5] Гельфонд А.О., Линник Ю.В. Элементарные методы в аналитической теории чисел.-М.:ФМлит,1962.-272с.

[6] Уокер Р. Алгебраические кривые.-М.:ИЛ,1952.-236с.

[7] Прасолов В.В., Соловьев Ю.П. Эллиптические функции и алгебраические уравнения.-М.:Факториал,1997.-145с.

[8] EL Gamal T. A public key cryptosystem and a signature scheme based on discrete logarithms.//IEEE Transactions on Information Theory.-1985.-v.IT-31.-no.4.-p.473-481.

[9] Galbraith S.D., Smart N.P. A cryptographic application of Weil descent. Cryptography and Coding.// 7th IMA Conference, Springer-Verlag, LNCS-1999.-v.1746-p.191-200. (The full version of the paper is HP Labs Technical Report, HPL-1999-70.)

[10] Виноградов И.М. Теория чисел. М.:Наука,1990.-167с.

[11] Гашков С.Б., Чубариков В.Н. Арифметика. Алгоритмы. Сложность вычислений.-МГУ.М.:Дрофа,2005.-320с.

[12] Глухов М.М., Елизаров В.П., Нечаев А.А. Алгебра.-М.:Гелиос АРВ,т.1,2.2003.-416с.

[13] Коблиц Н. Курс теории чисел и криптографии.-М.:ТВП,2001.-269с.

[14] Лидл Р., Нидеррейтер Х. Конечные поля.т.1,2.-М.:Мир,1988.-818с.

[15] Нечаев В.И. Элементы криптографии.-М.:Высшая школа,1999.-112с.

[16] Jungnickel D. Finite fields: Structure and arifmetics.-Mannhein-Leipzig-Wien-Zurich, Wissenschaftsverlag,1993.

[17] Stinson D.R. Cryptography: Theory and practice.-London:CRC Press,1995.-434p.

[18] Riesel H. Some soluble cases of the discrete logarithm problem // BIT-1988.-v.28-n.4-p.839-851.

[19] Череннёв М.А. Криптографические протоколы.-М.: Изд-во центра прикладных исследований при механико-математическом факультете.-2006.-70с.

[20] Гашков С.Б., Применко Э.А., Черепнев М.А. Криптографические методы защиты информации. Учебное пособие-М.:Академия.-2010.-298с.

[21] Череннёв М.А. О некотором свойстве дискретного логарифма.// Тез. докл. XII международной конференции "Проблемы теоретической кибернетики Нижний новгород.-1999.-с.246.

[22] Rivest R., Shamir A., Adleman L.N. A method for obtaining digital signatures and public-key cryptosystems.// Communications of the ACM.-1978.- v.21-p.120-126.

[23] Черепнёв М.А. О связи сложностей задач дискретного логарифмирования и Диффи-Хеллмэна.// Дискретная математика.-1996.-т.8-вып.3-с.22-30.

[24] Pratt V. Every prime has a succinct certificate// SIAM J. Comput.-1975.-v.4-n.3-p.214-220

[25] Shoup V. Lower bounds for discrete logarithms and related problems.// Advances in Cryptology - EUROCRYPT'97.-Springer-Verlag.-LNCS.-1997.-v.1233-p.256-266.

[26] Maurer U.M., Wolf S. The Diffie-Hellman Protocol.// Designs, Codes and Cryptography, Special Issue Public Key Cryptography, Kluwer Academic Publishers-2000.-v.19-n.3-p.147-171.

[27] Черепнёв М.А. Обращение спариваний для решения задачи дискретного логарифмирования. Фундаментальная и прикладная математика, 2013, Т.18, Вып.4, стр.185-195.

[28] Ford K., Konyagin S.V., Luca F. Prime chains and Pratt trees.// GAFA-2010.-v.20-n.5-p.1231-1258.

[29] Черепнёв М.А. Некоторые свойства больших простых делителей чисел вида $p-1$.// Математические заметки-2006.-80:6- p.920-925

[30] Черепнёв М.А. Некоторые свойства больших простых делителей чисел вида $p-1$.// Материалы конференции МаБИТ 2003.-2003.-с.218-220.

[31] Черепнёв М.А. О величине простых делителей чисел вида $p-1$.//Материалы конференции МаБИТ 2004.-2004.-с.243-246.

[32] Canfield E.R., Erdos P., Pomerance C. On a problem of Oppenheim concerning "Factorisatio Numerorum"// J.Number Theory-1983.-v.17-p.1-28.

[33] Selivanov B.I. On waiting time in the scheme random allocation of coloured particles.// Discrete Math. Appl.-1995.-v.5(1)-p.73-82.

[34] Silverman J. The Arithmetic of Elliptic Curves.-Springer,1986.-513+xviiip.

[35] Сидельников В.М., Черепнёв М.А., Ященко В.В. Системы открытого распределения ключей на основе некоммутативных полугрупп.// ДАН СССР-1993.-т.332-№5-с.566-567.

[36] Черепнёв М.А. Схемы открытого распределения ключей на основе некоммутативной операции.// Тез. докл. Тез. докл. XIII международной конференции "Проблемы теоретической кибернетики Казань.-2002.-с.190.

[37] Черепнев М.А. Схемы открытого распределения ключей на основе некоммутативной группы.// Дискретная математика.-2003.-т.15-вып.2-с.47-51.

[38] Назаров В.В. Об использовании свойства коммутирования символа степенного вычета в схемах открытого распределения ключей. Диссертация на соискание учёной степени кандидата физико-математических наук. М.,2006.-90с.

[39] Раинчик В.С. Представления некоторых логарифмических функций в кольце целых алгебраических чисел многочленами малой степени.// Математические заметки.-2012.-т.92-вып.1-с.68-73.

[40] Конявская С.В. Электронные замки: проблема выбора. // Информационная безопасность.-2007.-№4-с.58-60.

[41] Панфилов Б.А., Черепнёв М.А., Панфилов Ю.Б. Электронные замки на основе смешанной системы счисления, реализованной с помощью резистивной матрицы памяти на базе полярнозависимого

электромассопереноса в кремнии.// Радиотехника и электроника.-2005.-т.50-№12-с.1523-1527.

[42] Панфилов Б.А., Черепнев М.А. Симметричная криптосистема шифрования на основе смешанной системы счисления.// Радиотехника и электроника.-2008.-т.53-№10-с.1314-1316.

[43] Патент на изобретение №2358082. Способ создания электронного кодового устройства повышенной криптографической стойкости (варианты)/Панфилов Б.А., Черепнёв М.А., Панфилов Ю.Б. заявка №2006123305, от 10.06.2009.;Приоритет 30.06.2006; Срок действия 30.06.2026

[44] M.E.Campbell On fixed points for discrete logsrithms. A thesis submitted in partial satisfaction of the requirements for the degree of Master of Arts in Mathematics.-University of California at Berkley.,2003.-47p.

[45] Черепнев М.А. Некоторые эффективные оценки для числа решений задачи Бризолиса.// Современные проблемы математики и механики.Математика.-2009.-т.IV-вып.3-с.125-129.

[46] R.K.Guy Unsolved Problems in Number Theory.(2-nd edition)-New York-Berlin, Springer-Verlag,1994.-303p.

[47] Cobeli C. and Zaharescu A. An exponential congruence with solutions in primitive roots.// Rev. Roumaine Math. Pures Appl.-1999.-v.44(1)-p.15-22.

[48] Исследования класса математических алгоритмов на эллиптических кривых в целях обеспечения достоверности документов и идентификации отправителей в системе межведомственного электронного документооборота. Отчёт о НИР (заключит.)/МГУ; Руководитель Ю.В.Нестеренко; Черепнёв М.А., Поповян И.А., Назаров В.В., Герман О.Н.-Шифр темы "Локон-Б"ГР № 2/15-40.-М.,2004.

[49] Кнут Д. Искусство программирования на ЭВМ. т.2.-С.-П.:Вильямс-2000.-682с.

[50] Основы криптографии (3-е издание)./Алферов А.П., Зубов А.Ю., Кузьмин А.С., Черемушкин А.В.;М.:Гелиос АРВ,2005.-480с.

[51] Черепнёв М.А. Замечание о ядре группового гомоморфизма метода спуска Вейля. Фундаментальная и прикладная математика 2014, Т.19, Вып.6, стр. 211-222

[52] Элементарное введение в эллиптическую криптографию. Алгебраические и алгоритмические основы./ А.А.Болотов, С.Б.Гашков, А.Б.Фролов, А.А.Часовских;М.:КомКнига,2006.-320с.

[53] Элементарное введение в эллиптическую криптографию. Протоколы криптографии на эллиптических кривых./ А.А.Болотов, С.Б.Гашков, А.Б.Фролов, А.А.Часовских;М.:КомКнига,2006.-272с.

[54] Хофман Л. Современные методы защиты информации.-М.:Сов.Радио,1980.-262с.

[55] Blake I.,Seroussi G., Smart N. Elliptic curves in cryptography.-London Mathematical Society 265, Cambridge University Press,1999.

[56] Blake I.,Seroussi G., Smart N. Advances in elliptic curve cryptograhy.-London Mathematical Society 317, Cambridge University Press,2005.

[57] Huesmoeller D. Elliptic Curves.-Springer-Verlag,2000.-510p.

[58] Washington L.C. Elliptic curves number theory and cryptography.-London,CRC Press,2003.

[59] Исследования класса математических алгоритмов на эллиптических кривых в целях обеспечения достоверности документов и идентификации

отправителей в системе межведомственного электронного документооборота. Отчёт о НИР (заключит.)/МГУ; Руководитель Ю.В.Нестеренко; Черепнёв М.А., Поповян И.А., Назаров В.В., Герман О.Н.-Шифр темы "Локон-А"ГР № 02/243.-М.,2002.-177с.

[60] N.Koblitz Algebraic Aspects of Cryptography.-Springer-Verlag,1998.-109p.

[61] Боревич З.И.,Шафаревич И.Р. Теория чисел.-М.:Наука,1985.-495с.

[62] Гекке Э. Лекции по теории алгебраических чисел.-М.:Гостехиздат,1940.-261с.

[63] Cassels J. W. S. Diophantine equations with special reference to elliptic curves. // J. LMS,-1966.-v.41-p.193-291.

[64] Weber H. Lehrbuch der Algebra. v.I, II,III.// New York, Chelsea Publishing Company,1902.-722p.

[65] Cohen H. A course in computational algebraic number theory.-Berlin,Heidelberg,Springer-Verlag.-1993.-545p.

[66] Wilkening J., Yu J. A local construction of the Smith normal form of a matrix polynomial. Journal of Symbolic Computation 46(2001), p.1-22.

[67] Storjohann A. Algorithms for matrix Canonical Forms. A dissertation for the degree of Doctor of Technical Sciences. 2013, 180p.

[68] S. D. Galbraith, F. Hess, N. P. Smart. Extending the GHS Weil descent attack. // Advances in Cryptology (EUROCRYPT 2002), Springer-Verlag, LNCS-2002.-v.2332-p.29-44.

[69] P. Gaudry, F. Hess, N. P. Smart. Constructive and destructive facets of Weil descent on elliptic curves. http://Hewlett Packard Laboratories Technical Report, 2000.-20p.(ultralix.polytechnique.fr/Labo/Pierrick.Gaudry/papers.html)

[70] A. Menezes, Qu M. Analysis of the Weil descent attack of Gaudry, Hess and Smart. // Topics in Cryptology - CT-RSA 2001, Springer-Verlag LNCS 2020, 308-318, 2001.

[71] Черепнев М.А. Вариант блочного алгоритма типа Ланцоша решения систем линейных уравнений.// Материалы Третьей международной научной конференции по проблемам безопасности и противодействия терроризму. МГУ 25-27.10.2007 (МаБИТ-2007)-М.:МЦНМО,2008.-с.129-136.

[72] Черепнев М.А. Блочный алгоритм типа Ланцоша решения разреженных систем линейных уравнений.// Дискр. Мат-ка-2008.-т.20-вып.1-с.145-150.

[73] Черепнев М.А. Алгоритмы построения матричных приближений Паде.// Материалы Четвертой международной научной конференции по проблемам безопасности и противодействия терроризму. МГУ 30-31.10.2008 (МаБИТ-2008)-М.:МЦНМО,2009.-с.21-22.

[74] Результаты внедрены в НПО " Квант" . Имеется акт внедрения.

[75] Черепнев М.А. О некоторых вычислениях в пространствах Крылова над $GF(2)$.// Вестник Тамбовского университета Сер. Естественные и технические науки.-2009.-т.14-вып.4-с.833-835.

[76] Исследование проблем теории чисел и алгебраической геометрии, связанных с анализом и синтезом шифрсистем : Отчет о НИР (заключит.)/Академия криптографии; Руководитель М.М.Глухов; М.А.Черепнев, Н.Л.Замарашкин и др.- Шифр темы "Идеал-3"; ГР №239-09.-М., 2010.-42с.

[77] Cherepniov M.A. Version of block Lanczos-type algorithm for solving sparse linear systems.// Bull.Math.Soc.Sci.Math.Roumanie.-2010.-v.53(101)-no.3-p.225-230. (http://rms.unibuc.ro/bulletin)

[78] Cherepniov M.A. Some estimations of performance of parallel algorithms for solving large linear systems over GF(2).// A Journal of Tambov State University, The works of participants of International conference "ParCA"presented according to the results of reviewing by International Program Commettee.-2010.-v.15-Iss.4-p.1342-1353.

[79] Результаты внедрены в ОАО "Концерн "Автоматика". Имеется акт внедрения.

[80] Черепнев М.А. Связь приближений ряда и базиса пространства Крылова в блочных алгоритмах Копперсмита и Монтгомери.// Фундаментальная и прикладная математика 2011/2012, том 17, выпуск 5, стр. 211-223

[81] Гашков С.Б., Гашков И.Б. Алгоритм Берлекемпа-Месси, цепные дроби, аппроксимации Паде и ортогональные многочлены.// Математические заметки.-2006.-т.79-вып.1-с.45-59.

[82] Черепнёв М.А. Обращение спариваний для решения задачи дискретного логарифмирования. Фундаментальная и прикладная математика (Journal of Mathematical Sciences), 2013, Т.18, Вып.4, стр.185-195. Journal of Mathematical Science: Volume 206, Issue 6 (2015), page 734-741.

[83] Федеральная целевая программа "Универсальный метод решения линейных систем над конечным полем на экзофлопных вычислителях": промежуточный отчёт "Выбор направлений исследований"/ ИВМ РАН, Руководитель Тыртышников Е.Е.; Черепнев М.А., Замарашкин Н.Л. и др. ГР № 2014-14-576-0054, 2014г. -91с.

[84] Черепнев М.А. Замечание о ядре группового гомоморфизма метода спуска Вейля. Фундаментальная и прикладная математика 2014, Т.19, Вып.6, стр. 211-222. Journal of Mathematical Sciences, Vol. 221, No. 3, March, 2017 p. 452-460

[85] Черепнев М.А.(совм. С Замарашкиным Н.Л.) Универсальный блочный метод Ланцоша-Паде для систем линейных уравнений над большими простыми полями. Фундаментальная и прикладная математика 2014, Т.19, Вып.6, стр. 223-247. Journal of Mathematical Sciences, Vol. 221, No. 3, March, 2017 p. 461-478

[86] Черепнев М.А. (совм. С Замарашкиным Н.Л.) Универсальный блочный метод Ланцоша-Паде для систем линейных уравнений над большими простыми полями. Труды международной конференции "Russian supercomputing days" (28-29 сентября 2015г. Москва), М., изд-во МГУ, стр. 509-520

[87] BFFblockLanzosPade свидетельство о регистрации прав на ПО. Авторы: Замарашкин Н.Л., Черепнев М.А., Желтков Д.А., Салуев Т.Г.,Тыртышников Е.Е. Номер: 2015662444. Дата получения: 24 ноября 2015 г.Описание: Параллельный алгоритм для решения линейных систем над большим простым полем - блочный вариант алгоритма Ланцоша-Паде.

[88] GF2blockLanzosPade свидетельство о регистрации прав на ПО. Авторы: Замарашкин Н.Л., Черепнев М.А., Желтков Д.А., Салуев Т.Г.,Тыртышников Е.Е. Номер: 2015662694. Дата получения: 30 ноября 2015 г. Описание: Параллельный блочный метода Ланцоша-Паде для систем линейных алгебраичеких уравнений над полем GF(2).

[89] Черепнев М.А. Обобщение алгоритма Риссанена на блочно-ганкелевы матрицы Дискретная математика, Т.28, вып.1, 2016 с.150-155.

[90] Черепнев М.А.Модулярный алгоритм приведения матриц к Смитовой нормальной форме. Дискретная математика Т.28, вып.2, 2016 с.160-164.

[91] Boer der B. Diffie-Hellman is as strong as discrete log for certain primes.// Proc. Crypto'88, Lect. Notes in Comp. Sc.-1989.-v.403-p.530-539.

[92] McCurley K.S. A key distribution system equivalent to fuctoring.// J. Cryptology.-1988.-v.1-№2-p.95-105.

[93] К.Прахар. Распределение простых чисел.-М.,Мир,1967.-511с.

[94] Maurer U.M. Towards the equivalence of breaking the Diffie-Hellman protocol and computing discrete logarithms.// Proc. Crypto'94.-1994.-p.271-281.

[95] Василенко О.Н. Теоретико-числовые алгоритмы в криптографии.-М.:МЦНМО,2006.-325с.

[96] Sakurai K., Shizuya H. Relationships among the Computational Powers of Breaking Diskrete Log Cryptosystems.// Eurocrypt'95,-1995.-p.341-355.

[97] Galbraith S.D., Raminder S.R. Using equivalence classes to accelerate solving the discrete logarithm problem in short interval.// PKC 2010, Springer, LNCS-2010.-v.6056-p.368-383.

[98] Hardy G.H., Ramanujan S. The normal number of prime factors of a number n.// Quart. J. of Math.,-1917.-v.48-p.76-92.

[99] Turán P. On a theorem of Hardy and Ramanujan.// J. Lond. math. Soc.-1929.-v.30-no2-p.93-111.

[100] Erdös P. On the normal number of prime factors of $p-1$ and some related problems concerning Euler's φ-function.// Quart. J. Oxford,-1935.-v.6-p.205-213.

[101] Canfield E.R., Erdos P., Pomerance C. On a problem of Oppenheim concerning "Factorisatio Numerorum"J.Number Theory 17(1983), 1-28.

[102] Selivanov B.I. On waiting time in the scheme random allocation of coloured particles. Discrete Math. Appl. 5(1), 73-82 (1995)

[103] Silverman J. The Arithmetic of Elliptic Curves.-Springer,1986.-513+xviiip.

[104] Galbraith S., Hess F., Vercauteren F. Aspects of pairing invertion. IEEE Trans., Dec. 2008, v.54, Issue:12, p.5719-5728.

[105] Canfield E.R., Erdos P., Pomerance C. On a problem of Oppenheim concerning "Factorisatio Numerorum"J.Number Theory 17(1983), 1-28.

[106] Selivanov B.I. On waiting time in the scheme random allocation of coloured particles. Discrete Math. Appl. 5(1), 73-82 (1995)

[107] Davenport J.H. On the integration of Algebraic Functions. LNCS 102 (1979), Springer-Verlag, Berlin

[108] Miller V.S. Short Programs for functions on Cyrves. Unpublished manuscript. 1986. http://crypto.stanford.edu/miller/

[109] Cohen H., Frey G. ets. Handbook of Elliptic and Hyperelliptic curve Cryptography. Chapman and Hall, London, New York, Singapore 2006.

[110] Ford K., Konyagin S.V., Luca F. Prime Chains and Pratt Trees, Geom. Funct. Anal., 20 (2010),. 1231–1258.

[111] Hess F. Pairing Lattices. In Pairing 2008, LNCS 5209, p. 18-38, Springer-Verlag, Berlin-Heidelberg-New York, 2008.

[112] Hess F. A Note on the Tate Pairing of Curves over Finite Fields. Arch. Math. (Basel)82 (2004), 28-32.

[113] Chang-An Zhao, Fangguo Zhang and Jiwu Huang A Note on the Ate Pairing Cryptology ePrint Archive: Report 2007/247. Cryptology ePrint Archive: Report 2007/247, http://eprint.iacr.org/2007/247.pdf

[114] Vasilenko O.N. Number-theoretic algorithms in Cryptography. 2006, 328c.

[115] Vercauteren F. The hidden root problem.In S.D.Galbraith, K.G.Paterson (eds.) Pairing based Cryptography - Pairing 2008, LNCS v.5209, pp.89-99. Springer, Heidelberg (2008)https://eprint.iacr.org/2008/261.pdf

[116] Vercauteren F. Optimal Pairings. IEEE Transactions on Information Theory, Volume 56 (1): 455-461, 2010. https://eprint.iacr.org/2008/096.pdf

[117] ван дер Варден Б.Л. Алгебра. М., Наука, 1976, 648 с.

[118] Gradshteyn I. S. and Ryzhik I. M. Tables of Integrals, Series, and Products, 6th ed. San Diego, CA: Academic Press, pp. 1111-1112, 2000.

[119] Faugere J.C., Din M.S., Verron T. On the Complexity of Computing Grobner Bases for Quast-Homogeneous Systems. ISSAC'13, June 26-29, 2013, Boston, Massachusetts, USA.

[120] Bardet M., Faugere J.C., Salvy B. Complexity of Grober basis computation for Semi-regular Overdetermined sequences over $\mathbb{F}_2$ with solutions in $\mathbb{F}_2$. INRIA, rapport de recherche № 5049, Decembre 2003.

[121] Дринфельд В.Г. Эллиптические модули.// Мат. сб.-1977.-т.102(144)-№2-р.594-627.

[122] Artin E., Tate J. Class field theory.-London-Amsterdam,Harvard,1961.-259p.

[123] Adleman L. M., Pomerance C., Rumely R. S. On Distinguishing prime numbers from composite numbers.// Annals of Mathematics.-1983.-v.117-p.173-206.

[124] Панфилов Б.А. // Труды LV научн. сессии посв. дню радио. М.: РНТОЭС им. А.С.Попова.-2000.-с.293.

[125] Ноден П., Китте К. Алгебраическая алгоритмика.-М.: Мир,1999.-719с.

[126] Davenport H. Multiplicative Number Theory. Second Edition.-New York,Springer-Verlag,-1980.-187p.

[127] Lang S. Elliptic curves:Diophantine analysis.-u.s.a., Springer-Verlag,1978.-265p.

[128] Koblitz N. Elliptic curve cryptosystems.// Math. Comp.-1987.-v.48-p.203-209.

[129] Зарисский О., Самюэль П. Коммутативная алгебра.-М.:Изд. ин. лит.-1963.-811р.

[130] Ленг С. Алгебраические числа.-М.:Мир.-1966.-221с.

[131] Cohen H. A course in computational algebraic number theory.-Berlin,Heidelberg,Springer-Verlag.-1993.-545p.

[132] Айерленд К., Роузен М. Классическое введение в современную теорию чисел.-М.:Мир,1987.-416с.

[133] Степанов С.А. Арифметика алгебраических кривых.-М.:Наука,1991.-368с.

[134] Galbraith S., Menezes A. Algebraic curves and cryptography.// Finite fields and their applications.-2005.-v.11-p.544-577.

[135] Коблиц Н. Введение в эллиптические кривые и модулярные формы.-М.:Мир,1988.-320с.

[136] Кнэпп.Э. Эллиптические кривые.-М.:Факториал Пресс,2004.-488с.

[137] Ленг С. Эллиптические функции.-М.:Наука,1984.-312с.

[138] Эллиптические кривые и современные алгоритмы теории чисел./Ю.П.Соловьев,В.А.Садовничий,Е.Т.Шавгулидзе,В.В.Белокуров; -Москва-Ижевск,-2003.-192с.

[139] Lercier R. Finding good random ellilptic curves for cryptosystems defined $GF(2^n)$.// Eurocript-97, Lecture Notes in Computer Science-1997.-v.1233-p.379-392.

[140] Шевалле К. Введение в теорию алгебраических функций от одной переменной.-М.:Физматгиз,-1959.-336с.

[141] Ленг С. Алгебра.-М.:Мир,-1968.-558с.

[142] Атья М., Макдональд И., Введение в коммутативную алгебру.-М.:Мир,-1972.-144с.

[143] Stichtenoth H. Algebraic function fields and codes.-Springer,-1998.-262c.

[144] Joux A., Lercier R. Improvements to the general number field sieve for discrete logarithms in prime fields.// Math. Comp.-2003.-v.72(242)-p.953-967.

[145] Montgomery P.L. A Block Lanczos Algorithm for Finding Dependencies over GF(2)// Advances in Cryptology - EuroCrypt'95 / Louis C.Guillou and Jean-Jacques Quisquater, editors. Berlin: Springer-Verlag, -Lect. Notes in Comp. Sci.-1995.-v.921-p.106-120.

[146] Coppersmith D. Solving homogeneous linear systems over GF(2) via block Wiedemann algorithm.// Mathematics of Computation.-1994-v.62-no.205-p.333-350.

[147] Kaltofen E. Analysis of Coppersmith's block Wiedemann algorithm for the parallel solution of sparse linear systems.// Math. Comp.-1995.-v.64(210)-p.777-806.

[148] Giorgi P., Jannerod C.-P., Villard G. On the complexity of polynomial matrix computations.// Proc. ISSAC'03,Philadelfia, USA.-2003.-p.1-8. (http://www.ens-lyon.fr)

[149] Jeannerod C.-P. and Villard G. Asymptotically fast polynomial matrix algorithms for multivariable systems.// Int. J. Control.-2006.-v.79(11)-p.1359- 1367.

[150] Ramajulu M. Parallel computations for the matrix stage of integer factorization.// A Project report submitted in partial fulfillment of the requirements for

the degree of Master of Technology in Computational science,-Bangalore,-2008.-62p.(http://www.serc.iisc.ernet.in)

[151] Алисейчук А.Н. Неасимптотические оценки вероятности распределения ранга случайной матрицы над конечным полем.// Дискретная математика.-2007.-т.19-№2-с.85-93.

[152] Astakhov V.V. Estimstes of the running time and memory requirements of the new algorithm of solving large sparse linear systems over the field with two elements.// A Journal of Tambov State University, The works of participants of International conference "ParCA"presented according to the results of reviewing by International Program Commettee.-2010.-v.15-Iss.4-p.1311-1327.

[153] LaMacchia B.A., Odlyzko A.M. Solving large sparse linear systems over finite fields.// Advances in Cryptology - Crypto'95, -LNCS-1995.-v.537-p.109-133.

[154] Eberly W. and Kaltofen E. On randomized Lanczos algorithms.// Proc. of the 1997 International Symposium on Symbolic Algebraic Computation, -ACM Press.-1997.-p.176-183.

[155] Global Combine on Mesh Architectures with Wormhole Routing./ Barnett M., Littlefeld R., Payne D.G., Geijn V.D.// Proc. of 7-th Int. Parallel Proc. Symp.-1993.

[156] Hovinen B. and Eberly W. A reliable block Lanczos algorithm over small finite fields.// Proceedings of the 2005 international Symposium on Symbolic and Algebraic Computation.-Beijing, -China, -July 24 - 27.-2005.-(www.cpsc.ucalgary.ca/ eberly/Research/publications.php.)

[157] Beckermann B. and Labahn G. A uniform approach for the fast computation of Matrix-type Pade approximants.// SIAM J. Matrix Analysis and Applications.-1994.-p.1-26.

[158] Результаты внедрены в НПО "Квант", 2014г. Акт внедрения прилагается.

[159] Об экономном построении транзитивного замыкания ориентированного графа./ В.Л.Арлазаров, Е.А.Диниц, М.Кронрод, И.А.Фараджев//ДАН СССР.-1970.-т.194-№3-с.487-488.

[160] Cantor D.G., Kaltofen E. On fast multiplication of polynomials over arbitrary Algebras.// Acta Informatica.-1991.-v.28-p.693-701.

[161] Thome E. Subquadratic computation of vector generating polynomials and improvement of the block Wiedemann algorithm.// Journal of Symbolic Computation.-2002.-v.33:5-p.757-775.

[162] Factorization of a 768-bit RSA modulus. version 1.0. 07.01./ Kleinjung T., Aoki K., Franke J., Lenstra A.K., Thomé E., Bos J.W., Gaudry P., Kruppa A., Montgomery P.L., Osvik D.A., Riele H., Timofeev A., Zimmermann P.-2010.-(http://eprint.iacr.org/2010/006.pdf)-34p.

[163] Результаты внедрены в ФГУП "НИИ Автоматики" в рамках ОКР ""65с17-2М", инв.№30857п, 2011г. Акт внедрения прилагается.

[164] Результаты внедрены в ОАО "Концерн "Автоматика" в рамках ОКР "65с17-2М", инв.№30857п, 2011г. Акт внедрения прилагается.

[165] Черепнев М.А. A connection of series approximations and the basis of the Krylov space in block algorithms of Coppersmith and Montgomery. Journal of Mathematical Sciences: -Volume 193, -Issue 4 (2013), -p.622-630.

[166] Coppersmith D. Fast evaluation of logarithms in fields of characteristic two.// IEEE Trans.Inform.Theory.-1984.-v.30-p.587-594.

[167] Lanczos C. An iteration method for the solution of the eigenvalue problem of linear differential and integral operators.// J. Res. Nat. Bur. Standards, -45 (1950), -p.255–282

[168] Coppersmith D. Solving linear equations over $GF(2)$: Block Lanczos algorithm. Linear Algebra Appl., -193:33-60, -1993.

[169] Gutknecht M.H. A completed theory of the unsymmetric Lanczos process and related algorithms. Part I. // SIAM J. Matrix Anal. Appl., -13(2):594-639, -1992.

[170] Gutknecht M.H. A completed theory of the unsymmetric Lanczos process and related algorithms. Part II.// SIAM J. Matrix Anal. Appl., -15:15-58, -1994.

[171] Montgomery P. A block Lanczos algorithm for finding dependencies over GF(2), volume 921.// Springer-Verlag, 1995.

[172] Peterson M. and Monico C. $\mathbb{F}_2$ Lanczos revisited. // Linear Algebra Appl., -428:1135-1150, -2008.

[173] Дорофеев А.Я. Вычисление логарифмов в конечном простом поле методом линейного решета.// Труды по дискретной математике, -5:29-50, -2002.

[174] Дорофеев А.Я. Решение систем линейных уравнений при вычислении логарифмов в конечном простом поле.// Математические вопросы криптографии, -3(1):5-51, -2012.

[175] Поповян И.А., Нестеренко Ю.В., Гречников Е.А. Вычислительно сложные задачи теории чисел.// Учебное пособие.-Издательство Московского Университета, -2012.

[176] Замарашкин Н.Л. Алгоритмы для разреженных систем линейных уравнений в GF(2).// Учебное пособие. -Издательство Московского Университета, -2013.